Petrol Bunk Management System

A PROJECT REPORT

Submitted by

JAYAVIGNESH KSD (1010108012)

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE 641 049

(An autonomous institution affiliated to Anna University, Chennai)

APRIL 2014

Petrol Bunk Management System

A PROJECT REPORT

Submitted by

JAYAVIGNESH KSD (1010108012)

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE 641 049

(An autonomous institution affiliated to Anna University, Chennai)

APRIL 2014

KUMARAGURU COLLEGE OF TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this project report "**PETROL BUNK MANAGEMENT SYSTEM**" is the bonafide work of **JAYAVIGNESH KSD (10101080012)**who carried out the project work under my supervision.

SIGNATURE

Associate Prof.P.Devaki

SUPERVISOR

Associate Professor
Department of Computer Science
Kumaraguru College of Technology
Coimbatore 641049

SIGNATURE

Prof. N.JAYAPATHI

HEAD OF THE DEPARTMENT

Professor and Head
Department of Computer Science
Kumaraguru College of Technology
Coimbatore 641049

The candidate with University Register number 1010108012 was examined in the project viva-voce examination held on

INTERNAL EXAMINER

Name:

Designation:

EXTERNAL EXAMINER

Name:

Designation:

ACKNOWLEDGEMENT

We express our hearty gratitude to our beloved Chairmen of our Institution,**Padmabhushan Arutselvar Dr .N. Mahalingam, B.Sc., F.I.E.,**for giving us this great opportunity to pursue this course.

We thank **Dr. R.S.Kumar, Ph.D.,** Principal, Kumaraguru College of Technology, Coimbatore, for being a constant source of inspiration and providing us with the necessary facilities to work on this project.

I would like to make special knowledgement and thanks to **Prof.N.Jayapathi, M.Tech,** Head of Department of Computer Science and Engineering, for his support and encouragement throughout the project.

I express our deep gratitude and gratefulness to my project coordinator and my guide**Ms. P. Devaki, M.E., M.S.,** Associate Professor, Department of Computer Science and Engineering, for her supervision, enduring patience, active involvement and guidance.

We articulate our thankfulness to our class advisor, **Mr. K.Sivan ArulSelvan, M.E., Associate Professor**, Department of Computer Science and Engineering. We thank him for support, encouragement and ideas. We also thank for the countless hours that he has spent with us, and seamless support for our academic choices.

We also express our profound gratitude to our parents, petrol bunk owner and friends for their support.

ABSTRACT

The petrol bunk management system is developed for the sole purpose helping bunk owner manage their petrol bunk in an efficient way. As of date petrol bunk owners used log books and manually record entries of the pump discharge and they use a slip known as tank lorry slip for noting down the fuel loads being delivered to the bunk through means of tanker lorry. It is a mandate specified by oil corporations to follow this procedure. Moreover they also maintain several others logs which make it a tedious process to follow. This management system reduces the number of data entry points thus reducing a major part of this tedious process.

The admin console allows the admin to add a user or remove a user and similarly add and remove credit parties. The admin can provide access to a trusted person of the bunk to take care of the task they have to perform. Moreover it is possible for the admin to view or track the credit parties' bills. The admin can also print the two pages as a monthly report which changes to a simple printable format when print function is called.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO.	ABBREVIATION	DESCRIPTION
1.	AJAX	Asynchronous JavaScript and XML
2.	CSS	Cascading Style Sheets
3.	PHP	Hypertext Pre-processor
4.	SQL	Structured Query Language
5.	MS	Motor Sprits
6.	HSD	High Speed Diesel
7.	DBMS	Database Management System
8.	PBMS	Petrol Bunk Management System

CHAPTER 1

INTRODUCTION

The petrol bunk management system is designed to efficiently manage the petrol bunks and with ease. This application acts as easy alternative to tedious manual data entry with multiple entry points and calculations was made manually. The petrol bunk management system simplifies the work the user just by getting data from the user and then performing calculations thus saving the user a lot of time and a faster management system. The petrol bunk management can be easily managed using the admin console. Users can be added and removed as per need of admin and also track customers. Generally though management of bunks seem quite easy there are several task that are to be done

1.1EXISTING SYSTEM

The existing system used in the petrol bunk is manual entry methods and involves no automation or any computerized methods. The bunk owner has to manually fill in the details. Three log books are maintained which are the density register, meter reading register and customer bill book. Every day the bunk owner or one of his staff employees have to manually fill these record books and then make calculation which takes on an average of 1 hour. There are multiple data entry points and data might be entered wrongly and this might cause a fatal flaw in calculation resulting in uncalculated loss or profit. Moreover if the client wants to view any particular data of a particular month it will be a tedious process to search for that particular data.

1.2 PROPOSED SYSTEM

The development of this new system contains the following facilities

- Computerized entry for density register, meter reading.
- A PDF is generated for the tanker lorry slip which can be then be printed and stuck on the tanker lorry
- A login page is provided to authenticate users. Based on authentication access is provided. Only the admin can access the admin panel and other users are not granted access.
- An admin panel with 5 functionalities are provided
  - Check customer credit
  - Add new user
  - Remove user
  - Add new Creditbillparty
  - Remove Creditbillparty

1.2 IMPLEMENTATION

The work is implemented using mysql DBMS and Apache web server. Implementation of this project is done with a single database. For the evaluation purpose, a sample of 10 records is inserted in the database. The GUI provides the user with the option of choosing what operation to perform. There are various operations like adding credit party details, add or remove a user, print details and pdf generation. The results of the operation are displayed in user interface and the changes are reflected in the respective table in the database. The login interface is given to the users to enter the user id and password. The authentication details are verified using user table that is available in the database.

CHAPTER 2

LITERATURE REVIEW

While virtual management environments have been available in some capacity since 1960, “the PLATO system featured multiple roles, including students who could study assigned lessons and communicate with teachers through on-line notes, instructors, who could examine student progress data, as well as communicate and take lessons themselves, and authors, who could do all of the above, plus create new lessons” (Wikipedia, 2006a, 1960s section,). Management systems have only been available, in roughly their present form, since the 1990s (Vollmer, 2003), with Blackboard and WebCT being broadly adopted in universities and colleges by early 2000 (Online, 2006). Initial versions of an LMS focused on organizing and managing course content and learners. As with many organizations, higher education was unsure about the role of technology in the educational process. The rapid penetration of management systems as key tools for management occurs in a vacuum of solid research as to their effectiveness in increasing management—or even indication of best practices for technology implementation. Pedagogy is generally a secondary consideration to student management; some researchers attempted to bridge research from face-to-face environments to technology spaces (Chickering & Ehrmann, 1996)—a practice that may be convenient, but errs in assuming that the online space is an extension of physical instruction, not an alternative medium with unique affordances. Managementsystems became the default starting point of technology enabled management in an environment largely omitting faculty and learner needs.

CHAPTER 3

SYSTEM SPECFICATION

3.1 HARDWARE REQUIREMENTS

S.NO.	HARDWARE	REQUIREMENT
1	Processor Type	Intel Core i5
2	Processor Speed	2.53 GHz
3	RAM	4 GB RAM
4	Hard Disk Capacity	320 GB
5	Monitor	14/17”
6	Mouse	Optical
7	Keyboard	101/104 Keys

Table 3.1 Hardware Requirements

3.2 SOFTWARE REQUIREMENTS

- Text Editor
- Web browsers (latest version)
- Apache Web Server
- MYSQL DBMS
- Development Tools: PHP,CSS,HTML,SQL

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 FRONT END

4.1.1 HTML

HTML is the predominant mark-up language for web pages. HTML is the basic building-blocks of web pages. The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

Web browsers can also refer to Cascading Style Sheets(CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentation HTML mark-up.

4.1.2 PHP

Hypertext Pre-processor is a general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document.

As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as a standalone interpreter on most operating systems and computing platforms.

4.1.3 JQUERY

JQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. It is currently developed by a team of developers led by Dave Methvin. Used by over 80% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

JQuery is free, open source software, licensed under the MIT License. JQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. JQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

The set of jQuery core features — DOM element selections, traversal and manipulation — enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM-data-structures; and influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX framework and

ASP.NET MVC Framework while Nokia has integrated it into the Web Run-Time widget development platform.JQuery has also been used in MediaWiki since version 1.16

4.2 BACKEND

4.2.1 MYSQL

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MYSQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

CHAPTER 5

MODULES WITH DESCRIPTION

5.1 LOGIN

The login page is the first page the user sees when they open the website on their web browser. The login page is used to authenticate the users before they access the application. The admin has a higher privilege and is provided extra functionality such as adding new user or removing users and similarly for credit bill party and he check on the amount the credit party has to pay. The login page is coded to prevent any SQL injections.

5.2 PDF GENERATOR

The tanker lorry slip needs to be printed 8 times for the petrol and 12 times for diesel as per the regulations set by Hindustan petroleum. So the tank lorry slip form receives the data that needs to be entered in the PDF. Once the form is submitted the data is then sent to a PHP file which then generates the PDF file using the library using FPDF.

5.3 DENSITY REGISTER

The log book format for density register has been replicated in created the density register form. It is a liveform where the user can double click on a cell and enter data as well he change data by double clicking on a cell. These changes are updated to the database instantly after they have been verified. The user even can take a print out of a month for the current year as a form. The form automatically adds dates into the table if the month is found to be missing. The year changes after Dec 31st. This form gets it invoice no from the tank lorry slip thus reducing the number of data entry points. The density register automatically converts the density to 15°C.

5.4 METER READING

The meter reading formis similar to density register. There are 2 tanks in total. One is a petrol tank and the other is a diesel tank. The petrol tank is connected to four pumps and the diesel tank is connected to four pumps. The meter reading of these pumps is stored in these forms. This is a live form and data can be updated while viewing the data by double clicking. A monthly report can be taken as a print if needed and generates dates automatically.

5.5 CREDITBILLPARTY

The credit bill party is a dynamic form where the credit customer bills are entered. In this dynamic form the more no of items the more no of rows can be generated. Once the customers' bills are entered, admin tracks each customer.

5.6 ADMIN CONSOLE

The admin console is a page that can only be accessed if logged in as admin user. The admin console is page that can be used to perform functionalities. These functionalities are used to define control of users of application. It consists of five different functionalities. They are:

- **Check customer credit**

  Here each customer can be tracked and his credit can be checked
- **Add new user**

  A new user can be added to add data to these forms
- **Remove user**

  Any user can be removed by the admin.
- **Add credit party**

  A new credit party can be added
- **Remove credit party user**

  Any credit party user can be removed.

CHAPTER 6

TESTING AND IMPLEMENTATION

6.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good testing case is one that has a high probability of finding an as-yet undiscovered error. A successful test is one that uncovers an as-yet undiscovered error. Testing demonstrates that software appears to be working according to specification and that performance requirement appears to have been met.

The following list of categories shows that this software has been tested.

- No bugs block the execution of tests.
- Distinct output is generated for each input.
- The product evolves in the functional stages (allows simultaneous development and testing).
- Internal errors are automatically reported.
- Source code is accessible.
- All possible outputs can be generated through some combination of inputs.
- All code is executed through some combination of inputs.
- Tests can be conveniently specified, automated and reproduced.
- Software modules can be tested independently.
- Structural simplicity (architecture is modularized to limit the propagation of faults).
- Changes to the software are infrequent.
- Design to software is well understood.
- Changes to the design are communicated.

Software testing is often referred as verification and validation. Verification refers to the set of activities that ensures that the software is implemented correctly.

6.1.1TYPES OF TESTING

- System testing.
- Unit testing
- Integration testing
- Validation testing
- Output testing
- User acceptance testing

6.1.1.1 SYSTEM TESTING

Testing of the debugging programs is one of the most criticalaspects of the computer programming triggers, without programs that works, the system world never produce the output for which it was designed. Testing is best performed when user development are asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quality of the data used the matter of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commends.

6.1.1.2 UNIT TESTING

Unit testing tests each module individually and integrate with the overall system. Unit testing focuses on the verification efforts in the smallest unit of software design. This is also known as module testing which is carried out during programming stage itself. In unit testing, each module is found to working satisfactorily as regard to the expected output from the module. There are same validation checks for fields also. It is very easy to find error debut in the system.

6.1.1.3 INTEGRATION TESTING

Data can be lost across an interface; one module can have an adverse effort on the other sub function when combined may not produces the desired major function. Integrated testing is the systematic testing for constructing the uncover errors which merging different modules of a software. The need for integrated test is to find the overall system performance.

6.1.1.4 VALIDATION TESTING

Validation testing can be defined many ways but a simple definition is that validation succeeds when the software function in manner that can be reasonably expected be the customer. After validation test has been conducted one of the two possible conditions exists. The function or performance characteristics confirm to specification and one accepted.

6.1.1.5 OUTPUT TESTING

Output testing is the one without which no system could be useful if it does not produce the required output the specific format. Asking the user about the format required by system tests the output displayed or generated by the system consideration.

The output format is considered the of screen display. The output format on the screen is fund to be correct as the format was designed in the system phase according to the user need. For the hard copy also the output comes out as specified by the user. Hence the output testing does not result in any correction in the system.

6.1.1.6 USER ACCEPTANCE TESTING

The software is deployed to the end users and tested for all the possible test cases from their view of requirements. The user acceptance testing is an important factor, during the testing of software.

6.2SYSTEM IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not a guarantee improvement in the organization using the new system, as well as, important installation will prevent any improvement

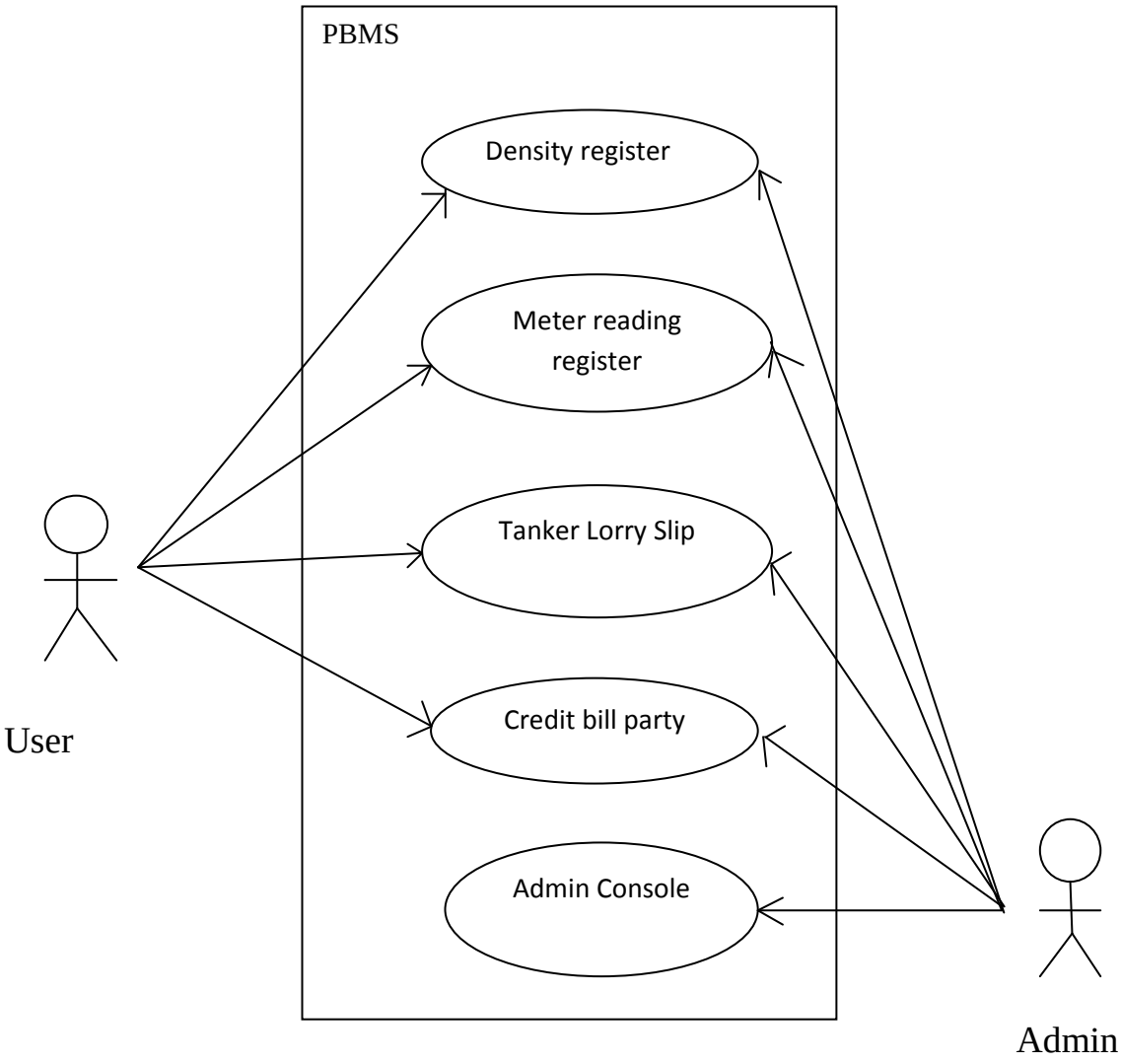
The implementation phase involves the following tasks.

- Careful planning
- Investigation of system and constraints
- Design of methods to achieve the Changeover
- Training of staff in the Changeover phase

CHAPTER 7

DIAGRAM

7.1 USECASE DIAGRAM



CHAPTER 8

CONCLUSION

Thus through the use of the petrol bunk management system the admin is able to manage his petrol bunk over the LAN and is also able to monitor the bunk remotely. The time spent on manually calculating the log registers is now managed by the management system and system being a webapplication the user is able to host the application elsewhere and use it from a different place.

Further advancements in this software would be automatically obtain the readings from the pump and mail notification to those clients whose credit value goes beyond a particular limit and asking them to within a particular deadline. Providing a login space for the credit party where could make a order online should that when his/her truck arrives the order would be ready even if there was no stock while placing the order. Automatic calculation of sales from the pumps. Automating most of the process. The only task of the admin would be to enter the load that has arrived and even the bill could be directly entered into the system.

A.1 SOURCE CODE:

Index.php

```
<?php
error_reporting(0);
session_start();
if(isset($_SESSION["sessionid"]) &&
$_SESSION["sessionid"]=="true")
 header('Location: welcome.php');
try
{
 if(!empty($_POST))
 {
  $con=mysqli_connect("localhost","root","","project") or
die("Error");
  if(isset($_POST["username"]) && isset($_POST["password"]))
  {
   $user=mysql_real_escape_string($_POST["username"]);
   $password=mysql_real_escape_string($_POST["password"]);
   $query_result=mysqli_query($con,"SELECT * FROM Login
WHERE username='$user' AND password='$password';");
   if(($query_result->num_rows)>0)
   {
```

```
    $_SESSION["username"]=$_POST["username"];
    $_SESSION["sessionid"] = "true";
    header('Location: welcome.php');
   }
   else
    throw new Exception("Invalid credentials");
  }
  else
   throw new Exception("Refresh page");
  mysqli_close($con);
 }
}
catch(Exception $e)
{
 $error=$e->getMessage();
}
?>
<html>
<head>
<style>
body
```

```
{
 background-color:#0056aa;
}
label
{
 color:#FFFFFF;
 font-size:16;
}
.login
{
 background-color:#3288DC;
 color:#FFFFFF;
 position:relative;
 width:13%;
 top:40%;
 left:40%;
 border:2px solid;
 border-top-left-radius:20px;
 border-top-right-radius:20px;
 border-bottom-left-radius:20px;
 border-bottom-right-radius:20px;
```

```
}
.Heading
{
 color:#FFFFFF;
 position:relative;
 top:40%%;
 left:45%;
}
</style>
<script src="js/jquery-latest.js"></script>
</head>
<body>
<div class="Heading"><h1>Login</h1></div>
<div class="login">
<form method="POST">
<table width="100%">
<tr>
<td><label for="username">Username:</label></td><td><input type="text" id="username" name="username"></td>
</tr>
<tr>
<td><label for="password">Password:</label></td><td><input type="password" id="password" name="password"></td>
```

```
</tr>
<tr>
<td colspan="2" align="center"><input type="image"
src="images/login.png" align="center"></td>
</tr>
<tr>
<td colspan="2" align="center"><label id="Error"><?php echo
$error; ?></label></td>
</tr>
</div>
</body>
</html>
```

Welcome.php

```
<?php
error_reporting(0);
session_start();
if(!isset($_SESSION["sessionid"]) &&
$_SESSION["sessionid"]!="true")
 header('Location: index.php');
 if($_GET["logout"]=="true")
 {
  session_destroy();
```

```
  header('Location: index.php');
 }
?>
<!DOCTYPE HTML>
<html>
<head><title></title>
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="main">
<div class="logout"><a
href="welcome.php?logout=true"><input type="image"
src="images/logoutbutton.png"></a></div>
<div class="username" sytle="position:fixed;"><label>Welcome
<?php echo $_SESSION["username"]; ?></label></div>
<div class="heading"><h1><a href="welcome.php">Petrol Bunk
Management Software<br /></a></h1></div>
<div class="container">
<div class="menu_nav">
<ul class="links">
<li><a href="welcome.php"><span>Home</span></a></li>
<li><a href="denreg.php"><span>Density
Register</span></a></li>
```

**
Meter reading
**

**
Tank Lorry Slip
**

**
Credit Bill entry
**

**<?php if($_SESSION["username"]=="Admin") echo'
Admin console
'?>**

**
**

** **

** **

** **

** **

** **

** **

</body>

</html>

denreg.php

<?php

error_reporting(0);

session_start();

```
if(!isset($_SESSION["sessionid"]) &&
$_SESSION["sessionid"]!="true")
 header('Location: index.php');
if($_GET["logout"]=="true")
{
 session_destroy();
 header('Location: index.php');
}
if(isset($_POST['tank']))
{
  $tank=$_POST['tank'];
}
else
{
  $tank="MS16KL";
}
$con=mysqli_connect('localhost','root','','project');
if($tank=="MS16KL")
  $quantity="MS_Quantity";
else
  $quantity="HSD_Quantity";
if($_POST['month']!=NULL)
```

```
                                                    $month=
$_POST['month'];
else
                                                    $month=
"01";
$year=date("Y");
$date=$year.'-'.$month.'-'."__";
if($_POST['tank']!=NULL)
                                                    $tank=$
_POST['tank'];

$result=mysqli_query($con,"SELECT * FROM denreg_$tank
WHERE date LIKE '$date';");
mysqli_close($con);
?>

<html>
<head>
<title>Density Register</title>
<style>
@media screen
{
html
{
```

```
 padding:0;
 margin:0;
 background-color:#0056aa;
 color: #FFFFFF;
}
select
{
 color: #FFFFFF;
 background-color:#3288DC;
 border-color:#3288DC;
}
a
{
 text-decoration:none;
 color: #FFFFFF;
}
a:visited
{
 color:#FFFFFF;
}
a:hover
{
```

```
 color:black;
}
.heading
{
 text-align:center;
}
.logout
{
 position:absolute;
 right:10%;
 top:1%
}
{
 list-style-type: none;
}
ul li
{
 display: inline;
 padding:6px;
}
.menu_nav_back
{
```

```
 position:fixed;
 width:100%;
 height:56px;
 top:80px;
 background-color:white;
 z-index: 1;
}
.menu_nav
{
 display:inline-block;
 background-color:#3288DC;
 border:2px solid;
 position:fixed;
 top:80px;
 left:35%;
 border-radius:8px;
 text-align:middle;
 z-index: 2;
}
.logout input[type="image"]
{
 width:20%;
```

```
}
.username
{
 position:absolute;
 left:10%;
 top:5%;
 font-size:22px;
}
td
{
 text-align:center;
}
#table1
{
 border-collapse:collapse;
 border:1px dotted #D0D0D0;
 background: url("images/blackbackground.jpg");
}
.data:hover
{
 background:#808080;
}
```

```
.content
{
 background-color:#0056aa;
 position:relative;
 left:1px;
 top:80px;
}
.clr
{
 height:100px;
 clear:both;
 padding:0;
 margin:0;
 width:100%;
 font-size:0px;
 line-height:0px;
}
}
@media print
{
 #table1
 {
```

```
 border-collapse:collapse;
 border:1px solid;
}
.logout,.username,.heading,.container
{
 display:none;
}
</style>
<script src="js/jquery-latest.js"></script>
<script type="text/javascript">
$('document').ready(function(){
var date=new Date();
var month=$('#monthval').val();
var decant=$('#tankval').val();
var year=date.getFullYear();
$('#year').text(year);
if(decant!=null)
$('#tank').val(decant);
if(month!=null)
$('#month').val(month);
$()
$('.data').attr('title','Double-click');
```

```
});
function change(value,sno)
{
 var day=$('td:first', $(value).parents('tr')).text();
 var month=$('#monthval').val();
 var date=new Date();
 var tank=$('#tankval').val();
 var year=date.getFullYear();
 date=(year+'-'+month+'-'+day);
 var data=prompt("Please enter new data",$(value).text());
 if(data!=null && data!="null")
 {
  $(value).text(data);
  $.ajax({
  type:"POST",
  url:"updatedenreg.php",
  data:{"todate":date,
  "serial":sno,"input":data,"tank":tank},
  success:function(data1){if(data1=="success")
  {
    $("form").submit();
  }
```

```
  else
   alert(data1);
  },
  error:function(){alert("Failed due to certain reasons");}
  });
 }
}
</script>
</head>
<body>
<div class="main">
<div class="logout"><a
href="welcome.php?logout=true"><input type="image"
src="images/logoutbutton.png"></a></div>
<div class="username" sytle="position:fixed;"><label>Welcome
<?php echo $_SESSION["username"]; ?></label></div>
<div class="heading"><h1><a href="welcome.php">Petrol Bunk
Management Software<br /></a></h1></div>
<div class="container">
<div class="menu_nav">
<ul class="links">
<li><a href="welcome.php"><span>Home</span></a></li>
<li><a href="denreg.php"><span>Density
Register</span></a></li>
```

Meter reading

Tank Lorry Slip

Credit Bill entry

<?php if($_SESSION["username"]=="Admin") echo'
Admin console
'?>

<form method="post">

<table>

<tr>

<td>Tank:</td><td><select id="tank" name="tank" onchange="this.form.submit();"><option value="MS16KL">MS16KL</option><option value="HSD22KL">HSD22KL</option></select></td>

<td>Month:</td><td><select id="month" name="month" onchange="this.form.submit();">

<option value="01">Jan</option>

<option value="02">Feb</option>

<option value="03">March</option>

<option value="04">April</option>

<option value="05">May</option>

<option value="06">June</option>

<option value="07">July</option>

<option value="08">August</option>

<option value="09">September</option>

<option value="10">October</option>

<option value="11">November</option>

<option value="12">December</option>

</select></td>

<td>Year:</td><td><label id="year"></td>

</tr>

</table>

</form>

<input type="hidden" id="monthval" value="<?php echo $month; ?>">

<input type="hidden" id="tankval" value="<?php echo $tank; ?>">

<table border="1" align="center" width="100%" id="table1">

<tr>

<td></td>

<td colspan="3">Morning Density</td>

<td colspan="4">Receipts</td>

```
</tr>
<tr>
<td></td>
<td colspan="2">Observed</td>
<td></td>
<td></td>
<td></td>
<td colspan="2">Observed</td>
</tr>
<tr>
<td>DATE</td>
<td>DENSITY</td>
<td>TEMPERATURE</td>
<td>DENSITY CONVERTED TO 15C</td>
<td>INVOICE NO</td>
<td>QUANTITY</td>
<td>DENSITY</td>
<td>TEMPERATURE</td>
</tr>
<?php
$con=mysqli_connect('localhost','root','','project');
if($result->num_rows==0)
```

```
{
 $num = cal_days_in_month(CAL_GREGORIAN,$month,$year);


 for($i=1;$i<=$num;$i++)
 {
  if($i<10)
   $datetemp=$year.'-'.$month.'-'.'0'.$i;
  else
   $datetemp=$year.'-'.$month.'-'.$i;
  mysqli_query($con,"INSERT INTO denreg_$tank(`Date`)
VALUES('$datetemp');") or die("Error");
 }
 $result=mysqli_query($con,"SELECT * FROM denreg_$tank
WHERE date LIKE '$date';");
}
 while($row = mysqli_fetch_array($result))
  {
   $date = strtotime($row["Date"]);
   $datesample=$row["Date"];
   $basic_data=mysqli_query($con,"SELECT
Invoice_No,$quantity FROM basic WHERE
Date='$datesample';");
   $basicrow=mysqli_fetch_array($basic_data);
   echo "<tr><td>";
```

```
  echo date("d",$date);
  echo '</td><td class=data ondblclick=change(this,1);>';
  echo $row['Morning_density'];
  echo '</td><td class=data ondblclick=change(this,2);>';
  echo $row['Morning_temp'];
  echo '</td><td class=15c>';
  echo '</td><td class=invoice>';
  echo $basicrow['Invoice_No'];
  echo '</td><td class=quantity>';
  echo $basicrow[$quantity];
  echo '</td><td class=data ondblclick=change(this,3);>';
  echo $row['Receipt_density'];
  echo '</td><td class=data ondblclick=change(this,4);>';
  echo $row['Receipt_temperature'];
  echo '</td></tr>';
 }
 mysqli_close($con);
?>
</table>
</div>
<div class="clr"></div>
</div>
```

```
</html>

Updatedenreg.php
<?php
$sno=$_POST['serial'];
$date=$_POST['todate'];
$data=$_POST['input'];
$tank=$_POST['tank'];
if($sno=="1")
{
                                                    $field="
Morning_density";
}
else if($sno=="2")
{
                                                    $field="
Morning_temp";
}
else if($sno=="3")
{
                                                    $field="
Receipt_density";
}
```

```
else if($sno=="4")
{
                                                    $field="
Receipt_temperature";
}
$con = mysqli_connect('localhost','root','','project') or
die("Error");
mysqli_query($con,"UPDATE denreg_$tank SET $field='$data'
WHERE Date='$date'") or die("Error");
echo "success";
// Check connection
if (mysqli_connect_errno())
{
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
mysqli_close($con);
?>
```

Meter_reading.php

```
<?php
error_reporting(0);
session_start();
if(!isset($_SESSION["sessionid"]) &&
$_SESSION["sessionid"]!="true")
 header('Location: index.php');
```

```
if($_GET["logout"]=="true")
{
 session_destroy();
 header('Location: index.php');
}
if($_POST['tank']!=NULL)
{
  $tank=$_POST['tank'];
}
else
{
  $tank="MS16KL";
}

if($_POST['month']!=NULL)
                                                    $month=
$_POST['month'];
else
                                                    $month=
"01";
$year=date("Y");
$date=$year.'-'.$month.'-'.'__';
```

```
$con=mysqli_connect('localhost','root','','project') or
die("Error");

$result=mysqli_query($con,"SELECT * FROM
meter_reading_$tank WHERE date LIKE '$date';")or
die("Error");

mysqli_close($con);

?>

<html>

<head></head>

<style>

@media screen

{

html

{

 padding:0;

 margin:0;

 background-color:#0056aa;

 color: #FFFFFF;

}

select

{

 color: #FFFFFF;

 background-color:#3288DC;
```

```
 border-color:#3288DC;
}
a
{
 text-decoration:none;
 color: #FFFFFF;
}
a:visited
{
 color:#FFFFFF;
}
a:hover
{
 color:black;
}
.heading
{
 text-align:center;
}
.logout
{
 position:absolute;
```

```
 right:10%;
 top:1%
}
ul
{
 list-style-type: none;
}
ul li
{
 display: inline;
 padding:6px;
}
.menu_nav_back
{
 position:fixed;
 width:100%;
 height:56px;
 top:80px;
 background-color:white;
 z-index: 1;
}
.menu_nav
```

```
{
 display:inline-block;
 background-color:#3288DC;
 border:2px solid;
 position:fixed;
 top:80px;
 left:35%;
 border-radius:8px;
 text-align:middle;
 z-index: 2;
}
.logout input[type="image"]
{
 width:20%;
}
.username
{
 position:absolute;
 left:10%;
 top:5%;
 font-size:22px;
}
```

```
td
{
                                                                text-
align:center;
}
.data:hover
{
 background:#808080;
}
.content
{
 background-color:#0056aa;
 position:relative;
 left:1px;
 top:80px;
}
#table1
{
 border-collapse:collapse;
 border:1px dotted #D0D0D0;
 background: url("images/blackbackground.jpg");
}
.clr
```

```
{
 height:100px;
 clear:both;
 padding:0;
 margin:0;
 width:100%;
 font-size:0px;
 line-height:0px;
}
}
@media print
{
 #table1
 {
  border-collapse:collapse;
  border:1px solid;
 }
 .logout,.username,.heading,.container
 {
  display:none;
 }
}
```

```
</style>
<script src="js/jquery-latest.js"></script>
<script type="text/javascript">
$('document').ready(function(){
var date=new Date();
var month=$('#monthval').val();
var decant=$('#tankval').val();
var year=date.getFullYear();
$('#year').text(year);
if(decant!=null)
$('#tank').val(decant);
if(month!=null)
$('#month').val(month);
});
function change(value,sno)
{
 var day=$('td:first', $(value).parents('tr')).text();
 var month=$('#monthval').val();
 var date=new Date();
 var tank=$('#tankval').val();
 var year=date.getFullYear();
 date=(year+'-'+month+'-'+day);
```

```
 var data=prompt("Please enter new data",$(value).text());
 if(data!=null && data!="null")
 {
  $(value).text(data);
  $.ajax({
  type:"POST",
  url:"updatemeterreg.php",
  data:{"todate":date,"serial":sno,"input":data,"tank":tank},
  success:function(data1){if(data1=="success")
  {
    $("form").submit();
  }
  },
  error:function(){alert(2);}
  });
 }
}
</script>
<body>
<div class="main">
<div class="logout"><a
href="welcome.php?logout=true"><input type="image"
src="images/logoutbutton.png"></a></div>
```

<label>Welcome <?php echo $_SESSION["username"]; ?></label>

<h1>Petrol Bunk Management Software
</h1>

<ul class="links">

Home

Density Register

Meter reading

Tank Lorry Slip

Credit Bill entry

<?php if($_SESSION["username"]=="Admin") echo'
Admin console
'?>

<form method="post">

<table>

```
<tr>
<td>Tank:</td><td><select id="tank" name="tank"
onchange="this.form.submit();">
<option value="MS16KL">MS16KL</option>
<option value="HSD22KL">HSD22KL</option>
</select></td>
<td>Month:</td><td><select id="month" name="month"
onchange="this.form.submit();">
<option value="01">Jan</option>
<option value="02">Feb</option>
<option value="03">March</option>
<option value="04">April</option>
<option value="05">May</option>
<option value="06">June</option>
<option value="07">July</option>
<option value="08">August</option>
<option value="09">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option>
</select></td>
<td>Year:</td><td><label id="year"></td>
</tr>
```

```
</table>

</form>

<input type="hidden" id="monthval" value="<?php echo $month; ?>">

<input type="hidden" id="tankval" value="<?php echo $tank; ?>">

<table border="1" id="table1">

<tr>

<td rowspan="2">Date</td><td colspan="4" align="center"><?php echo $tank; ?></td><td rowspan="2">Remarks</td>

</tr>

<tr>

<td>PRODUCT PUMP NO I</td>

<td>PRODUCT PUMP NO II</td>

<td>PRODUCT PUMP NO III</td>

<td>PRODUCT PUMP NO IV</td>

</tr>

<?php

$con=mysqli_connect('localhost','root','','project');

if($result->num_rows==0)

{

 $num = cal_days_in_month(CAL_GREGORIAN,$month,$year);

 for($i=1;$i<=$num;$i++)
```

```
 {
  if($i<10)
   $datetemp=$year.'-'.$month.'-'.'0'.$i;
  else
   $datetemp=$year.'-'.$month.'-'.$i;
  mysqli_query($con,"INSERT INTO meter_reading_$tank(`Date`) VALUES('$datetemp');") or die("Error");
 }
 $result=mysqli_query($con,"SELECT * FROM meter_reading_$tank WHERE date LIKE '$date';");
}
while($row=mysqli_fetch_array($result))
{
 $date=strtotime($row['Date']);
 echo "<tr><td>";
 echo date("d",$date);
 echo '</td><td class=data ondblclick=change(this,1);>';
 echo '</td><td class=data ondblclick=change(this,2);>';
 echo '</td><td class=data ondblclick=change(this,3);>';
 echo '</td><td class=data ondblclick=change(this,4);>';
 echo '</td><td class=data ondblclick=change(this,5);>';
 echo '</td></tr>';
```

```
}
?>
</table>
<div class="clr"></div>
</div>
</body>
</html>
```

FPDF.php

```
<?php
/*******************************************************************************
* FPDF                                                                         *
*                                                                              *
* Version: 1.7                                                                 *
* Date:    2011-06-18                                                          *
* Author:  Olivier PLATHEY                                                     *
*******************************************************************************/

define('FPDF_VERSION','1.7');

class FPDF
```

{

var $page; // current page number

var $n; // current object number

var $offsets; // array of object offsets

var $buffer; // buffer holding in-memory PDF

var $pages; // array containing pages

var $state; // current document state

var $compress; // compression flag

var $k; // scale factor (number of points in user unit)

var $DefOrientation; // default orientation

var $CurOrientation; // current orientation

var $StdPageSizes; // standard page sizes

var $DefPageSize; // default page size

var $CurPageSize; // current page size

var $PageSizes; // used for pages with non default sizes or orientations

var $wPt, $hPt; // dimensions of current page in points

var $w, $h; // dimensions of current page in user unit

var $lMargin; // left margin

var $tMargin; // top margin

var $rMargin; // right margin

var $bMargin; // page break margin

var $cMargin; // cell margin

var $x, $y; // current position in user unit

var $lasth; // height of last printed cell

var $LineWidth; // line width in user unit

var $fontpath; // path containing fonts

var $CoreFonts; // array of core font names

var $fonts; // array of used fonts

var $FontFiles; // array of font files

var $diffs; // array of encoding differences

var $FontFamily; // current font family

var $FontStyle; // current font style

var $underline; // underlining flag

var $CurrentFont; // current font info

var $FontSizePt; // current font size in points

var $FontSize; // current font size in user unit

var $DrawColor; // commands for drawing color

var $FillColor; // commands for filling color

var $TextColor; // commands for text color

var $ColorFlag; // indicates whether fill and text colors are different

var $ws; // word spacing

var $images; // array of used images

var $PageLinks; // array of links in pages

var $links; // array of internal links

```
var $AutoPageBreak;      // automatic page breaking
var $PageBreakTrigger;   // threshold used to trigger page breaks
var $InHeader;           // flag set when processing header
var $InFooter;           // flag set when processing footer
var $ZoomMode;           // zoom display mode
var $LayoutMode;         // layout display mode
var $title;              // title
var $subject;            // subject
var $author;             // author
var $keywords;           // keywords
var $creator;            // creator
var $AliasNbPages;       // alias for total number of pages
var $PDFVersion;         // PDF version number

/*******************************************************************************
*                                                                              *
*                               Public methods                                 *
*                                                                              *
*******************************************************************************/
function FPDF($orientation='P', $unit='mm', $size='A4')
{
```

```
                                        // Some
checks
                                        $this-
>_dochecks();
                                        //
Initialization of properties
                                        $this-
>page = 0;
                                        $this->n
= 2;
                                        $this-
>buffer = '';
                                        $this-
>pages = array();
                                        $this-
>PageSizes = array();
                                        $this-
>state = 0;
                                        $this-
>fonts = array();
                                        $this-
>FontFiles = array();
                                        $this-
>diffs = array();
                                        $this-
>images = array();
```

```
                                                $this-
>links = array();
                                                $this-
>InHeader = false;
                                                $this-
>InFooter = false;
                                                $this-
>lasth = 0;
                                                $this-
>FontFamily = '';
                                                $this-
>FontStyle = '';
                                                $this-
>FontSizePt = 12;
                                                $this-
>underline = false;
                                                $this-
>DrawColor = '0 G';
                                                $this-
>FillColor = '0 g';
                                                $this-
>TextColor = '0 g';
                                                $this-
>ColorFlag = false;
                                                $this-
>ws = 0;
```

```
                                                        // Font
path
                                                        if(define
d('FPDF_FONTPATH'))
                                                        {
                                                         $this-
>fontpath = FPDF_FONTPATH;

                                                        if(substr
($this->fontpath,-1)!='/' && substr($this->fontpath,-1)!='\\')

                                                        $this-
>fontpath .= '/';
                                                        }
                                                        elseif(is_
dir(dirname(__FILE__).'/font'))
                                                         $this-
>fontpath = dirname(__FILE__).'/font/';
                                                        else
                                                         $this-
>fontpath = '';
                                                        // Core
fonts
                                                        $this-
>CoreFonts = array('courier', 'helvetica', 'times', 'symbol',
'zapfdingbats');
```

```
                                                    // Scale
factor
                                                    if($unit=
='pt')
                                                     $this-
>k = 1;
                                                    elseif($u
nit=='mm')
                                                     $this-
>k = 72/25.4;
                                                    elseif($u
nit=='cm')
                                                     $this-
>k = 72/2.54;
                                                    elseif($u
nit=='in')
                                                     $this-
>k = 72;
                                                    else
                                                     $this-
>Error('Incorrect unit: '.$unit);
                                                    // Page
sizes
                                                    $this-
>StdPageSizes = array('a3'=>array(841.89,1190.55),
'a4'=>array(595.28,841.89), 'a5'=>array(420.94,595.28),
```

```
'letter'=
>array(612,792), 'legal'=>array(612,1008));
$size =
$this->_getpagesize($size);
$this-
>DefPageSize = $size;
$this-
>CurPageSize = $size;
// Page
orientation
$orienta
tion = strtolower($orientation);
if($orien
tation=='p' || $orientation=='portrait')
{
 $this-
>DefOrientation = 'P';
 $this-
>w = $size[0];
 $this-
>h = $size[1];
}
elseif($o
rientation=='l' || $orientation=='landscape')
{
```

```
                $this->DefOrientation = 'L';
                $this->w = $size[1];
                $this->h = $size[0];
            }
            else
                $this->Error('Incorrect orientation: '.$orientation);
            $this->CurOrientation = $this->DefOrientation;
            $this->wPt = $this->w*$this->k;
            $this->hPt = $this->h*$this->k;
            // Page margins (1 cm)
            $margin = 28.35/$this->k;
            $this->SetMargins($margin,$margin);
            // Interior cell margin (1 mm)
            $this->cMargin = $margin/10;
```

```
                                                    // Line
width (0.2 mm)
                                                    $this-
>LineWidth = .567/$this->k;
                                                    //
Automatic page break
                                                    $this-
>SetAutoPageBreak(true,2*$margin);
                                                    //
Default display mode
                                                    $this-
>SetDisplayMode('default');
                                                    // Enable
compression
                                                    $this-
>SetCompression(true);
                                                    // Set
default PDF version number
                                                    $this-
>PDFVersion = '1.3';
}

function SetMargins($left, $top, $right=null)
{
                                                    // Set
left, top and right margins
```

```
                                                    $this-
>lMargin = $left;
                                                    $this-
>tMargin = $top;
                                                    if($right
===null)
                                                     $right =
$left;
                                                    $this-
>rMargin = $right;
}

function SetLeftMargin($margin)
{
                                                    // Set left
margin
                                                    $this-
>lMargin = $margin;
                                                    if($this-
>page>0 && $this->x<$margin)
                                                     $this-
>x = $margin;
}

function SetTopMargin($margin)
```

```
{
                                                    // Set top
margin
                                                    $this-
>tMargin = $margin;
}

function SetRightMargin($margin)
{
                                                    // Set
right margin
                                                    $this-
>rMargin = $margin;
}

function SetAutoPageBreak($auto, $margin=0)
{
                                                    // Set
auto page break mode and triggering margin
                                                    $this-
>AutoPageBreak = $auto;
                                                    $this-
>bMargin = $margin;
                                                    $this-
>PageBreakTrigger = $this->h-$margin;
```

```
}

function SetDisplayMode($zoom, $layout='default')
{
                                                    // Set
display mode in viewer
                                                    if($zoom
=='fullpage' || $zoom=='fullwidth' || $zoom=='real' ||
$zoom=='default' || !is_string($zoom))
                                                     $this-
>ZoomMode = $zoom;
                                                    else
                                                     $this-
>Error('Incorrect zoom display mode: '.$zoom);
                                                    if($layou
t=='single' || $layout=='continuous' || $layout=='two' ||
$layout=='default')
                                                     $this-
>LayoutMode = $layout;
                                                    else
                                                     $this-
>Error('Incorrect layout display mode: '.$layout);
}

function SetCompression($compress)
```

```
{
                                                    // Set
page compression
                                                    if(functi
on_exists('gzcompress'))
                                                     $this-
>compress = $compress;
                                                    else
                                                     $this-
>compress = false;
}

function SetTitle($title, $isUTF8=false)
{
                                                    // Title
of document
                                                    if($isUT
F8)
                                                     $title =
$this->_UTF8toUTF16($title);
                                                    $this-
>title = $title;
}

function SetSubject($subject, $isUTF8=false)
```

```
{
                                                        //
Subject of document
                                                        if($isUT
F8)
                                                        $subject
= $this->_UTF8toUTF16($subject);
                                                        $this-
>subject = $subject;
}

function SetAuthor($author, $isUTF8=false)
{
                                                        //
Author of document
                                                        if($isUT
F8)
                                                         $author
= $this->_UTF8toUTF16($author);
                                                        $this-
>author = $author;
}

function SetKeywords($keywords, $isUTF8=false)
```

```
{
                                                        //
Keywords of document
                                                        if($isUT
F8)
                                                        $keywor
ds = $this->_UTF8toUTF16($keywords);
                                                        $this-
>keywords = $keywords;
}

function SetCreator($creator, $isUTF8=false)
{
                                                        //
Creator of document
                                                        if($isUT
F8)
                                                        $creator
= $this->_UTF8toUTF16($creator);
                                                        $this-
>creator = $creator;
}

function AliasNbPages($alias='{nb}')
```

```
{
                                                    // Define
an alias for total number of pages
                                                    $this-
>AliasNbPages = $alias;
}

function Error($msg)
{
                                                    // Fatal
error
                                                    die('<b>
FPDF error:</b> '.$msg);
}

function Open()
{
                                                    // Begin
document
                                                    $this-
>state = 1;
}

function Close()
```

```
{
                                                    //
Terminate document
                                                    if($this-
>state==3)
                                                      return;
                                                    if($this-
>page==0)
                                                      $this-
>AddPage();
                                                    // Page
footer
                                                    $this-
>InFooter = true;
                                                    $this-
>Footer();
                                                    $this-
>InFooter = false;
                                                    // Close
page
                                                    $this-
>_endpage();
                                                    // Close
document
                                                    $this-
>_enddoc();
```

```
}

function AddPage($orientation='', $size='')
{
                                                    // Start a
new page
                                                    if($this-
>state==0)
                                                     $this-
>Open();
                                                    $family
= $this->FontFamily;
                                                    $style =
$this->FontStyle.($this->underline ? 'U' : '');
                                                    $fontsize
= $this->FontSizePt;
                                                    $lw =
$this->LineWidth;
                                                    $dc =
$this->DrawColor;
                                                    $fc =
$this->FillColor;
                                                    $tc =
$this->TextColor;
                                                    $cf =
$this->ColorFlag;
```

```
                                                    if($this-
>page>0)
                                                    {
                                                     // Page
footer
                                                     $this-
>InFooter = true;
                                                     $this-
>Footer();
                                                     $this-
>InFooter = false;
                                                     // Close
page
                                                     $this-
>_endpage();
                                                    }
                                                    // Start
new page
                                                    $this-
>_beginpage($orientation,$size);
                                                    // Set
line cap style to square
                                                    $this-
>_out('2 J');
                                                    // Set
line width
```

```
                                                                $this-
>LineWidth = $lw;
                                                                $this-
>_out(sprintf('%.2F w',$lw*$this->k));
                                                                // Set
font
                                                                if($famil
y)
                                                                 $this-
>SetFont($family,$style,$fontsize);
                                                                // Set
colors
                                                                $this-
>DrawColor = $dc;
                                                                if($dc!='
0 G')
                                                                 $this-
>_out($dc);
                                                                $this-
>FillColor = $fc;
                                                                if($fc!='
0 g')
                                                                 $this-
>_out($fc);
                                                                $this-
>TextColor = $tc;
```

```
                                                    $this-
>ColorFlag = $cf;
                                                    // Page
header
                                                    $this-
>InHeader = true;
                                                    $this-
>Header();
                                                    $this-
>InHeader = false;
                                                    //
Restore line width
                                                    if($this-
>LineWidth!=$lw)
                                                    {
                                                     $this-
>LineWidth = $lw;
                                                     $this-
>_out(sprintf('%.2F w',$lw*$this->k));
                                                    }
                                                    //
Restore font
                                                    if($famil
y)
                                                     $this-
>SetFont($family,$style,$fontsize);
```

```
                                                                    //
Restore colors
                                                                    if($this-
>DrawColor!=$dc)
                                                                    {
                                                                     $this-
>DrawColor = $dc;
                                                                     $this-
>_out($dc);
                                                                    }
                                                                    if($this-
>FillColor!=$fc)
                                                                    {
                                                                     $this-
>FillColor = $fc;
                                                                     $this-
>_out($fc);
                                                                    }
                                                                    $this-
>TextColor = $tc;
                                                                    $this-
>ColorFlag = $cf;
}

function Header()
```

```
{
                                                    // To be
implemented in your own inherited class
}

function Footer()
{
                                                    // To be
implemented in your own inherited class
}

function PageNo()
{
                                                    // Get
current page number
                                                    return
$this->page;
}

function SetDrawColor($r, $g=null, $b=null)
{
                                                    // Set
color for all stroking operations
```

```
                                                    if(($r==0
&& $g==0 && $b==0) || $g===null)
                                                     $this-
>DrawColor = sprintf('%.3F G',$r/255);
                                                    else
                                                     $this-
>DrawColor = sprintf('%.3F %.3F %.3F
RG',$r/255,$g/255,$b/255);
                                                    if($this-
>page>0)
                                                     $this-
>_out($this->DrawColor);
}

function SetFillColor($r, $g=null, $b=null)
{
                                                    // Set
color for all filling operations
                                                    if(($r==0
&& $g==0 && $b==0) || $g===null)
                                                     $this-
>FillColor = sprintf('%.3F g',$r/255);
                                                    else
                                                     $this-
>FillColor = sprintf('%.3F %.3F %.3F rg',$r/255,$g/255,$b/255);
```

```
                                                    $this-
>ColorFlag = ($this->FillColor!=$this->TextColor);
                                                    if($this-
>page>0)
                                                     $this-
>_out($this->FillColor);
}

function SetTextColor($r, $g=null, $b=null)
{
                                                    // Set
color for text
                                                    if(($r==0
&& $g==0 && $b==0) || $g===null)
                                                     $this-
>TextColor = sprintf('%.3F g',$r/255);
                                                    else
                                                     $this-
>TextColor = sprintf('%.3F %.3F %.3F
rg',$r/255,$g/255,$b/255);
                                                    $this-
>ColorFlag = ($this->FillColor!=$this->TextColor);
}

function GetStringWidth($s)
```

```
{
                                                    // Get
width of a string in the current font
                                                    $s =
(string)$s;
                                                    $cw =
&$this->CurrentFont['cw'];
                                                    $w = 0;
                                                    $l =
strlen($s);
                                                    for($i=0;
$i<$l;$i++)
                                                     $w +=
$cw[$s[$i]];
                                                    return
$w*$this->FontSize/1000;
}

function SetLineWidth($width)
{
                                                    // Set
line width
                                                    $this-
>LineWidth = $width;
                                                    if($this-
>page>0)
```

```
                                                    $this->_out(sprintf('%.2F w',$width*$this->k));
}

function Line($x1, $y1, $x2, $y2)
{
                                                    // Draw a line
                                                    $this->_out(sprintf('%.2F %.2F m %.2F %.2F l S',$x1*$this->k,($this->h-$y1)*$this->k,$x2*$this->k,($this->h-$y2)*$this->k));
}

function Rect($x, $y, $w, $h, $style='')
{
                                                    // Draw a rectangle
                                                    if($style=='F')
                                                        $op = 'f';
                                                    elseif($style=='FD' || $style=='DF')
                                                        $op = 'B';
```

```
                                                    else
                                                     $op =
'S';
                                                    $this-
>_out(sprintf('%.2F %.2F %.2F %.2F re %s',$x*$this->k,($this-
>h-$y)*$this->k,$w*$this->k,-$h*$this->k,$op));
}

function AddFont($family, $style='', $file='')
{
                                                    // Add a
TrueType, OpenType or Type1 font
                                                    $family
= strtolower($family);
                                                    if($file=
='')
                                                     $file =
str_replace(' ','',$family).strtolower($style).'.php';
                                                    $style =
strtoupper($style);
                                                    if($style
=='IB')
                                                     $style =
'BI';
                                                    $fontkey
= $family.$style;
```

```
if(isset($this->fonts[$fontkey]))
  return;
$info = $this->_loadfont($file);
$info['i'] = count($this->fonts)+1;
if(!empty($info['diff']))
{
  //Search existing encodings
  $n = array_search($info['diff'],$this->diffs);
  if(!$n)
  {
    $n = count($this->diffs)+1;
    $this->diffs[$n] = $info['diff'];
  }
  $info['diffn'] = $n;
}
```

```
                                                    if(!empt
y($info['file']))
                                                    {
                                                     //
Embedded font
                                                    if($info['
type']=='TrueType')
                                                    $this-
>FontFiles[$info['file']] = array('length1'=>$info['originalsize']);
                                                     else
                                                    $this-
>FontFiles[$info['file']] = array('length1'=>$info['size1'],
'length2'=>$info['size2']);
                                                    }
                                                    $this-
>fonts[$fontkey] = $info;
}

function SetFont($family, $style='', $size=0)
{
                                                    // Select
a font; size given in points
```

```
                                                    if($famil
y=='')
                                                     $family
= $this->FontFamily;
                                                    else
                                                     $family
= strtolower($family);
                                                    $style =
strtoupper($style);
                                                    if(strpos
($style,'U')!==false)
                                                    {
                                                     $this-
>underline = true;
                                                     $style =
str_replace('U','',$style);
                                                    }
                                                    else
                                                     $this-
>underline = false;
                                                    if($style
=='IB')
                                                     $style =
'BI';
                                                    if($size=
=0)
```

```
                                                    $size =
$this->FontSizePt;
                                                    // Test if
font is already selected
                                                    if($this-
>FontFamily==$family && $this->FontStyle==$style && $this-
>FontSizePt==$size)
                                                     return;
                                                    // Test if
font is already loaded
                                                    $fontkey
= $family.$style;
                                                    if(!isset(
$this->fonts[$fontkey]))
                                                    {
                                                     // Test
if one of the core fonts

                                                    if($famil
y=='arial')

                                                    $family
= 'helvetica';

                                                    if(in_arr
ay($family,$this->CoreFonts))
                                                     {
```

```
if($family=='symbol' || $family=='zapfdingbats')
$style = '';
$fontkey = $family.$style;
if(!isset($this->fonts[$fontkey]))
$this->AddFont($family,$style);
  }
  else
$this->Error('Undefined font: '.$family.' '.$style);
}
// Select it
$this->FontFamily = $family;
$this->FontStyle = $style;
```

```
                                                    $this-
>FontSizePt = $size;
                                                    $this-
>FontSize = $size/$this->k;
                                                    $this-
>CurrentFont = &$this->fonts[$fontkey];
                                                    if($this-
>page>0)
                                                     $this-
>_out(sprintf('BT /F%d %.2F Tf ET',$this-
>CurrentFont['i'],$this->FontSizePt));
}

function SetFontSize($size)
{
                                                    // Set
font size in points
                                                    if($this-
>FontSizePt==$size)
                                                     return;
                                                    $this-
>FontSizePt = $size;
                                                    $this-
>FontSize = $size/$this->k;
                                                    if($this-
>page>0)
```

```
                                                    $this-
>_out(sprintf('BT /F%d %.2F Tf ET',$this-
>CurrentFont['i'],$this->FontSizePt));
}

function AddLink()
{
                                                    // Create
a new internal link
                                                    $n =
count($this->links)+1;
                                                    $this-
>links[$n] = array(0, 0);
                                                    return
$n;
}

function SetLink($link, $y=0, $page=-1)
{
                                                    // Set
destination of internal link
                                                    if($y==-
1)
                                                     $y =
$this->y;
```

```
                                                    if($page
==-1)
                                                     $page =
$this->page;
                                                    $this-
>links[$link] = array($page, $y);
}

function Link($x, $y, $w, $h, $link)
{
                                                    // Put a
link on the page
                                                    $this-
>PageLinks[$this->page][] = array($x*$this->k, $this->hPt-
$y*$this->k, $w*$this->k, $h*$this->k, $link);
}

function Text($x, $y, $txt)
{
                                                    //
Output a string
                                                    $s =
sprintf('BT %.2F %.2F Td (%s) Tj ET',$x*$this->k,($this->h-
$y)*$this->k,$this->_escape($txt));
                                                    if($this-
>underline && $txt!='')
```

```
                                                    $s .= '
'.$this->_dounderline($x,$y,$txt);
                                                    if($this-
>ColorFlag)
                                                    $s = 'q
'.$this->TextColor.' '.$s.' Q';
                                                    $this-
>_out($s);
}

function AcceptPageBreak()
{
                                                    // Accept
automatic page break or not
                                                    return
$this->AutoPageBreak;
}

function Cell($w, $h=0, $txt='', $border=0, $ln=0, $align='',
$fill=false, $link='')
{
                                                    //
Output a cell
                                                    $k =
$this->k;
```

```
                                                    if($this->y+$h>$this->PageBreakTrigger && !$this->InHeader && !$this->InFooter && $this->AcceptPageBreak())
                                                    {
                                                     //Automatic page break
                                                     $x = $this->x;
                                                     $ws = $this->ws;

                                                    if($ws>0)
                                                     {

                                                    $this->ws = 0;

                                                    $this->_out('0 Tw');
                                                     }
                                                     $this->AddPage($this->CurOrientation,$this->CurPageSize);
                                                     $this->x = $x;
```

```
                                                    if($ws>0
)
                                                     {
                                                    $this-
>ws = $ws;
                                                    $this-
>_out(sprintf('%.3F Tw',$ws*$k));
                                                     }
                                                    }
                                                    if($w==0
)
                                                     $w =
$this->w-$this->rMargin-$this->x;
                                                    $s = '';
                                                    if($fill ||
$border==1)
                                                    {
                                                     if($fill)
                                                    $op =
($border==1) ? 'B' : 'f';
                                                     else
```

```
                                                    $op =
'S';
                                                     $s =
sprintf('%.2F %.2F %.2F %.2F re %s ',$this->x*$k,($this->h-
$this->y)*$k,$w*$k,-$h*$k,$op);
                                                    }
                                                    if(is_stri
ng($border))
                                                    {
                                                     $x =
$this->x;
                                                     $y =
$this->y;

                                                    if(strpos
($border,'L')!==false)
                                                        $s
.= sprintf('%.2F %.2F m %.2F %.2F l S ',$x*$k,($this->h-
$y)*$k,$x*$k,($this->h-($y+$h))*$k);

                                                    if(strpos
($border,'T')!==false)
                                                        $s
.= sprintf('%.2F %.2F m %.2F %.2F l S ',$x*$k,($this->h-
$y)*$k,($x+$w)*$k,($this->h-$y)*$k);
```

```
                                                    if(strpos
($border,'R')!==false)
                                                        $s
.= sprintf('%.2F %.2F m %.2F %.2F l S ',($x+$w)*$k,($this->h-
$y)*$k,($x+$w)*$k,($this->h-($y+$h))*$k);

                                                    if(strpos
($border,'B')!==false)
                                                        $s
.= sprintf('%.2F %.2F m %.2F %.2F l S ',$x*$k,($this->h-
($y+$h))*$k,($x+$w)*$k,($this->h-($y+$h))*$k);
                                                    }
                                                    if($txt!=
='')
                                                    {

                                                    if($align
=='R')

                                                    $dx =
$w-$this->cMargin-$this->GetStringWidth($txt);

                                                    elseif($al
ign=='C')

                                                    $dx =
($w-$this->GetStringWidth($txt))/2;
```

```
                else
                    $dx = $this->cMargin;
                if($this->ColorFlag)
                    $s .= 'q '.$this->TextColor.' ';
                $txt2 = str_replace(')','\\)',str_replace('(','\\(',str_replace('\\','\\\\',$txt)));
                $s .= sprintf('BT %.2F %.2F Td (%s) Tj ET',($this->x+$dx)*$k,($this->h-($this->y+.5*$h+.3*$this->FontSize))*$k,$txt2);
                if($this->underline)
                    $s .= ' '.$this->_dounderline($this->x+$dx,$this->y+.5*$h+.3*$this->FontSize,$txt);
                if($this->ColorFlag)
                    $s .= ' Q';
                if($link)
                    $this->Link($this->x+$dx,$this->y+.5*$h-.5*$this->FontSize,$this->GetStringWidth($txt),$this->FontSize,$link);
```

```
                                                    }
                                                    if($s)
                                                     $this-
>_out($s);
                                                    $this-
>lasth = $h;
                                                    if($ln>0)
                                                    {
                                                     // Go to
next line
                                                     $this-
>y += $h;
                                                    if($ln==
1)
                                                    $this->x
= $this->lMargin;
                                                    }
                                                    else
                                                     $this-
>x += $w;
}

function MultiCell($w, $h, $txt, $border=0, $align='J', $fill=false)
```

```
{
                                                                    //
Output text with automatic or explicit line breaks
                                                                    $cw =
&$this->CurrentFont['cw'];
                                                                    if($w==0
)
                                                                     $w =
$this->w-$this->rMargin-$this->x;
                                                                    $wmax =
($w-2*$this->cMargin)*1000/$this->FontSize;
                                                                    $s =
str_replace("\r",'',$txt);
                                                                    $nb =
strlen($s);
                                                                    if($nb>0
&& $s[$nb-1]=="\n")
                                                                     $nb--;
                                                                    $b = 0;
                                                                    if($bord
er)
                                                                    {
                                                                    if($bord
er==1)
                                                                     {
```

```
                $border = 'LTRB';
                $b = 'LRT';
                $b2 = 'LR';
            }
            else
            {
                $b2 = '';
                if(strpos($border,'L')!==false)
                    $b2 .= 'L';
                if(strpos($border,'R')!==false)
                    $b2 .= 'R';
                $b = (strpos($border,'T')!==false) ? $b2.'T' : $b2;
```

```
  }
}
$sep = -1;
$i = 0;
$j = 0;
$l = 0;
$ns = 0;
$nl = 1;
while($i<$nb)
{
  // Get next character
  $c = $s[$i];
  if($c=="\n")
  {
        // Explicit line break
    if($this->ws>0)
```

```
                {
                    $this->ws = 0;
                    $this->_out('0 Tw');
                }
                $this->Cell($w,$h,substr($s,$j,$i-$j),$b,2,$align,$fill);
                $i++;
                $sep = -1;
                $j = $i;
                $l = 0;
                $ns = 0;
                $nl++;
                if($border && $nl==2)
```

```
                                                $b =
$b2;

                                                continue
;
                                                 }
                                                 if($c=='
')
                                                 {

                                                $sep =
$i;
                                                        $ls
= $l;

                                                $ns++;
                                                 }
                                                 $l +=
$cw[$c];

                                                if($l>$w
max)
                                                 {
                                                        //
Automatic line break
```

```
                if($sep==-1)
                    {
                if($i==$j)
                  $i++;
                if($this->ws>0)
                {
                  $this->ws = 0;
                  $this->_out('0 Tw');
                }
                $this->Cell($w,$h,substr($s,$j,$i-$j),$b,2,$align,$fill);
                    }
```

```
                else
                    {
                if($align
=='J')
                {
                 $this-
>ws = ($ns>1) ? ($wmax-$ls)/1000*$this->FontSize/($ns-1) : 0;
                 $this-
>_out(sprintf('%.3F Tw',$this->ws*$this->k));
                }
                $this-
>Cell($w,$h,substr($s,$j,$sep-$j),$b,2,$align,$fill);
                $i =
$sep+1;
                    }
                $sep = -
1;
```

```
                $j
= $i;
                $l
= 0;

            $ns = 0;

            $nl++;

            if($bord
er && $nl==2)

            $b =
$b2;

             }
             else

            $i++;
            }
            // Last
chunk
            if($this-
>ws>0)
            {
             $this-
>ws = 0;
```

```
                $this->_out('0 Tw');
            }
            if($border && strpos($border,'B')!==false)
                $b .= 'B';
            $this->Cell($w,$h,substr($s,$j,$i-$j),$b,2,$align,$fill);
            $this->x = $this->lMargin;
        }

        function Write($h, $txt, $link='')
        {
            // Output text in flowing mode
            $cw = &$this->CurrentFont['cw'];
            $w = $this->w-$this->rMargin-$this->x;
            $wmax = ($w-2*$this->cMargin)*1000/$this->FontSize;
            $s = str_replace("\r",'',$txt);
```

```
                                                $nb =
strlen($s);
                                                $sep = -
1;
                                                $i = 0;
                                                $j = 0;
                                                $l = 0;
                                                $nl = 1;
                                                while($i
<$nb)
                                                {
                                                 // Get
next character
                                                 $c =
$s[$i];
                                                if($c=="\
n")
                                                 {
                                                        //
Explicit line break
                                                $this-
>Cell($w,$h,substr($s,$j,$i-$j),0,2,'',0,$link);
                                                $i++;
```

```
                                                $sep = -
1;
                                                    $j
= $i;
                                                    $l
= 0;
                                                if($nl==
1)
                                                    {
                                                $this->x
= $this->lMargin;
                                                $w =
$this->w-$this->rMargin-$this->x;
                                                $wmax =
($w-2*$this->cMargin)*1000/$this->FontSize;
                                                    }
                                                $nl++;
                                                continue
;
                                                 }
```

```
                if($c==')
                $sep = $i;
                $l += $cw[$c];
                if($l>$wmax)
                {
                    // Automatic line break
                    if($sep==-1)
                    {
                        if($this->x>$this->lMargin)
                        {
                            // Move to next line
```

```
                                        $this->x = $this->lMargin;
                                        $this->y += $h;
                                        $w = $this->w-$this->rMargin-$this->x;
                                        $wmax = ($w-2*$this->cMargin)*1000/$this->FontSize;
                                        $i++;
                                        $nl++;
                                    continue;
                                    }
                                    if($i==$j)
                                        $i++;
```

```
                $this->Cell($w,$h,substr($s,$j,$i-$j),0,2,'',0,$link);
            }
            else
            {
                $this->Cell($w,$h,substr($s,$j,$sep-$j),0,2,'',0,$link);
                $i = $sep+1;
            }
            $sep = -1;
            $j = $i;
            $l = 0;
            if($nl==1)
            {
```

```
                                                    $this->x
= $this->lMargin;

                                                    $w =
$this->w-$this->rMargin-$this->x;

                                                    $wmax =
($w-2*$this->cMargin)*1000/$this->FontSize;
                                                        }

                                                    $nl++;
                                                     }
                                                     else

                                                    $i++;
                                                    }
                                                    // Last
chunk
                                                    if($i!=$j)
                                                     $this-
>Cell($l/1000*$this->FontSize,$h,substr($s,$j),0,0,'',0,$link);
}

function Ln($h=null)
{
```

```
                                                    // Line
feed; default value is last cell height
                                                    $this->x
= $this->lMargin;
                                                    if($h===
null)
                                                     $this-
>y += $this->lasth;
                                                    else
                                                     $this-
>y += $h;
}

function Image($file, $x=null, $y=null, $w=0, $h=0, $type='',
$link='')
{
                                                    // Put an
image on the page
                                                    if(!isset(
$this->images[$file]))
                                                    {
                                                     // First
use of this image, get info

                                                    if($type=
='')
```

```
{
$pos =
strrpos($file,'.');
if(!$pos)
$this-
>Error('Image file has no extension and no type was specified:
'.$file);
$type =
substr($file,$pos+1);
}
$type =
strtolower($type);
if($type=
='jpeg')
$type =
'jpg';
$mtd =
'_parse'.$type;
if(!meth
od_exists($this,$mtd))
```

```
                $this->Error('Unsupported image type: '.$type);
            $info = $this->$mtd($file);
            $info['i'] = count($this->images)+1;
            $this->images[$file] = $info;
        }
        else
            $info = $this->images[$file];

        // Automatic width and height calculation if needed
        if($w==0 && $h==0)
        {
            // Put image at 96 dpi
            $w = -96;
            $h = -96;
```

```
                                                                    }
                                                                    if($w<0)
                                                                     $w = -
$info['w']*72/$w/$this->k;
                                                                    if($h<0)
                                                                     $h = -
$info['h']*72/$h/$this->k;
                                                                    if($w==0
)
                                                                     $w =
$h*$info['w']/$info['h'];
                                                                    if($h==0
)
                                                                     $h =
$w*$info['h']/$info['w'];

                                                                    //
Flowing mode
                                                                    if($y===
null)
                                                                    {
                                                                     if($this-
>y+$h>$this->PageBreakTrigger && !$this->InHeader &&
!$this->InFooter && $this->AcceptPageBreak())
                                                                     {
```

```
                                                                //
Automatic page break

                                                                $x2 =
$this->x;

                                                                $this-
>AddPage($this->CurOrientation,$this->CurPageSize);

                                                                $this->x
= $x2;
                                                                 }
                                                                 $y =
$this->y;
                                                                 $this-
>y += $h;
                                                                }


                                                                if($x===
null)
                                                                 $x =
$this->x;
                                                                $this-
>_out(sprintf('q %.2F 0 0 %.2F %.2F %.2F cm /I%d Do
Q',$w*$this->k,$h*$this->k,$x*$this->k,($this->h-($y+$h))*$this-
>k,$info['i']));
                                                                if($link)
```

```
                                        $this->Link($x,$y,$w,$h,$link);
}

function GetX()
{
                                        // Get x position
                                        return $this->x;
}

function SetX($x)
{
                                        // Set x position
                                        if($x>=0)
                                        $this->x = $x;
                                        else
                                        $this->x = $this->w+$x;
}
```

```
function GetY()
{
                                                    // Get y
position
                                                    return
$this->y;
}

function SetY($y)
{
                                                    // Set y
position and reset x
                                                    $this->x
= $this->lMargin;
                                                    if($y>=0
)
                                                     $this-
>y = $y;
                                                    else
                                                     $this-
>y = $this->h+$y;
}

function SetXY($x, $y)
{
```

```
                                                    // Set x
and y positions
                                                    $this-
>SetY($y);
                                                    $this-
>SetX($x);
}

function Output($name='', $dest='')
{
                                                    //
Output PDF to some destination
                                                    if($this-
>state<3)
                                                     $this-
>Close();
                                                    $dest =
strtoupper($dest);
                                                    if($dest=
='')
                                                    {

                                                    if($name
=='')
                                                     {
```

```
$name =
'doc.pdf';

$dest =
'I';

 }
 else

$dest =
'F';

}
switch($
dest)

{
 case 'I':
      //
Send to standard output

$this-
>_checkoutput();

if(PHP_
SAPI!='cli')

      {
```

```
                                                    // We
send to a browser

                                                    header('
Content-Type: application/pdf');

                                                    header('
Content-Disposition: inline; filename="'.$name.'"');

                                                    header('
Cache-Control: private, max-age=0, must-revalidate');

                                                    header('
Pragma: public');
                                                        }

                                                    echo
$this->buffer;

                                                    break;
                                                     case
'D':
                                                        //
Download file

                                                    $this-
>_checkoutput();
```

```
                                                    header('
Content-Type: application/x-download');

                                                    header('
Content-Disposition: attachment; filename="'.$name.'"');

                                                    header('
Cache-Control: private, max-age=0, must-revalidate');

                                                    header('
Pragma: public');

                                                    echo
$this->buffer;

                                                    break;

                                                     case
'F':

                                                          //
Save to local file

                                                          $f
= fopen($name,'wb');

                                                    if(!$f)

                                                    $this-
>Error('Unable to create output file: '.$name);
```

```
                                                    fwrite($f
,$this->buffer,strlen($this->buffer));

                                                    fclose($f
);

                                                    break;
                                                     case
'S':
                                                          //
Return as a string

                                                    return
$this->buffer;
                                                     default:

                                                    $this-
>Error('Incorrect output destination: '.$dest);
                                                    }
                                                    return
'';
}

/**********************************************************
************************
```

```
*                                                               *
*                    Protected methods                          *
*                                                               *
*****************************************************************
************************/
function _dochecks()
{
                                                        // Check
availability of %F
                                                        if(sprintf
('%.1F',1.0)!='1.0')
                                                         $this-
>Error('This version of PHP is not supported');
                                                        // Check
mbstring overloading
                                                        if(ini_get
('mbstring.func_overload') & 2)
                                                         $this-
>Error('mbstring overloading must be disabled');
                                                        // Ensure
runtime magic quotes are disabled
                                                        if(get_m
agic_quotes_runtime())

                                                        @set_m
agic_quotes_runtime(0);
```

```
}

function _checkoutput()
{
                                                    if(PHP_
SAPI!='cli')
                                                    {

                                                    if(heade
rs_sent($file,$line))

                                                    $this-
>Error("Some data has already been output, can't send PDF file
(output started at $file:$line)");
                                                    }
                                                    if(ob_get
_length())
                                                    {
                                                      // The
output buffer is not empty

                                                    if(preg_
match('/^(\xEF\xBB\xBF)?\s*$/',ob_get_contents()))
                                                     {
                                                           //
It contains only a UTF-8 BOM and/or whitespace, let's clean it
```

```
                                                    ob_clean
();
                                                     }
                                                     else
                                                    $this-
>Error("Some data has already been output, can't send PDF
file");
                                                    }
}

function _getpagesize($size)
{
                                                    if(is_stri
ng($size))
                                                    {
                                                     $size =
strtolower($size);
                                                    if(!isset(
$this->StdPageSizes[$size]))
                                                    $this-
>Error('Unknown page size: '.$size);
```

```
                $a =
$this->StdPageSizes[$size];
                return
array($a[0]/$this->k, $a[1]/$this->k);
            }
            else
            {
                if($size[0
]>$size[1])
                return
array($size[1], $size[0]);
                else
                return
$size;
            }
}

function _beginpage($orientation, $size)
{
            $this-
>page++;
            $this-
>pages[$this->page] = '';
```

```
                                                    $this-
>state = 2;
                                                    $this->x
= $this->lMargin;
                                                    $this->y
= $this->tMargin;
                                                    $this-
>FontFamily = '';
                                                    // Check
page size and orientation
                                                    if($orien
tation=='')
                                                    $orienta
tion = $this->DefOrientation;
                                                    else
                                                    $orienta
tion = strtoupper($orientation[0]);
                                                    if($size=
='')
                                                      $size =
$this->DefPageSize;
                                                    else
                                                      $size =
$this->_getpagesize($size);
```

```
                                                    if($orientation!=$this->CurOrientation || $size[0]!=$this->CurPageSize[0] || $size[1]!=$this->CurPageSize[1])
                                                    {
                                                     // New size or orientation
                                                    if($orientation=='P')
                                                     {
                                                    $this->w = $size[0];
                                                    $this->h = $size[1];
                                                     }
                                                     else
                                                     {
                                                    $this->w = $size[1];
                                                    $this->h = $size[0];
                                                     }
```

```
                                                    $this->wPt = $this->w*$this->k;
                                                    $this->hPt = $this->h*$this->k;
                                                    $this->PageBreakTrigger = $this->h-$this->bMargin;
                                                    $this->CurOrientation = $orientation;
                                                    $this->CurPageSize = $size;
                                                }
                                                if($orientation!=$this->DefOrientation || $size[0]!=$this->DefPageSize[0] || $size[1]!=$this->DefPageSize[1])
                                                    $this->PageSizes[$this->page] = array($this->wPt, $this->hPt);
}

function _endpage()
{
                                                $this->state = 1;
}

function _loadfont($font)
```

```
{
                                                    // Load a
font definition file from the font directory
                                                    include(
$this->fontpath.$font);
                                                    $a =
get_defined_vars();
                                                    if(!isset(
$a['name']))
                                                     $this-
>Error('Could not include font definition file');
                                                    return
$a;
}

function _escape($s)
{
                                                    // Escape
special characters in strings
                                                    $s =
str_replace('\\','\\\\',$s);
                                                    $s =
str_replace('(','\\(',$s);
                                                    $s =
str_replace(')','\\)',$s);
```

```
                                                    $s =
str_replace("\r",'\\r',$s);
                                                    return
$s;
}

function _textstring($s)
{
                                                    //
Format a text string
                                                    return
'('.$this->_escape($s).')';
}

function _UTF8toUTF16($s)
{
                                                    //
Convert UTF-8 to UTF-16BE with BOM
                                                    $res =
"\xFE\xFF";
                                                    $nb =
strlen($s);
                                                    $i = 0;
                                                    while($i
<$nb)
```

```
                                                    {
                                                     $c1 =
ord($s[$i++]);
                                                    if($c1>=
224)
                                                     {
                                                           //
3-byte character
                                                    $c2 =
ord($s[$i++]);
                                                    $c3 =
ord($s[$i++]);
                                                    $res .=
chr((($c1 & 0x0F)<<4) + (($c2 & 0x3C)>>2));
                                                    $res .=
chr((($c2 & 0x03)<<6) + ($c3 & 0x3F));
                                                     }
                                                    elseif($c
1>=192)
                                                     {
```

```
                                                    //
2-byte character
                                                    $c2 =
ord($s[$i++]);
                                                    $res .=
chr(($c1 & 0x1C)>>2);
                                                    $res .=
chr((($c1 & 0x03)<<6) + ($c2 & 0x3F));
                                                     }
                                                     else
                                                     {
                                                        //
Single-byte character
                                                    $res .=
"\0".chr($c1);
                                                     }
                                                    }
                                                    return
$res;
}

function _dounderline($x, $y, $txt)
```

```
{
                                                                    //
Underline text
                                                                    $up =
$this->CurrentFont['up'];
                                                                    $ut =
$this->CurrentFont['ut'];
                                                                    $w =
$this->GetStringWidth($txt)+$this->ws*substr_count($txt,' ');
                                                                    return
sprintf('%.2F %.2F %.2F %.2F re f',$x*$this->k,($this->h-($y-
$up/1000*$this->FontSize))*$this->k,$w*$this->k,-
$ut/1000*$this->FontSizePt);
}

function _parsejpg($file)
{
                                                                    //
Extract info from a JPEG file
                                                                    $a =
getimagesize($file);
                                                                    if(!$a)
                                                                      $this-
>Error('Missing or incorrect image file: '.$file);
                                                                    if($a[2]!
=2)
```

```
                $this->Error('Not a JPEG file: '.$file);
            if(!isset($a['channels']) || $a['channels']==3)
                $colspace = 'DeviceRGB';
            elseif($a['channels']==4)
                $colspace = 'DeviceCMYK';
            else
                $colspace = 'DeviceGray';
            $bpc = isset($a['bits']) ? $a['bits'] : 8;
            $data = file_get_contents($file);
            return array('w'=>$a[0], 'h'=>$a[1], 'cs'=>$colspace, 'bpc'=>$bpc, 'f'=>'DCTDecode', 'data'=>$data);
        }

        function _parsepng($file)
```

```
{
                                                                    //
Extract info from a PNG file
                                                                    $f =
fopen($file,'rb');
                                                                    if(!$f)
                                                                     $this-
>Error('Can\'t open image file: '.$file);
                                                                    $info =
$this->_parsepngstream($f,$file);
                                                                    fclose($f
);
                                                                    return
$info;
}

function _parsepngstream($f, $file)
{
                                                                    // Check
signature
                                                                    if($this-
>_readstream($f,8)!=chr(137).'PNG'.chr(13).chr(10).chr(26).chr(
10))
                                                                     $this-
>Error('Not a PNG file: '.$file);
```

```
                                                    // Read
header chunk
                                                    $this-
>_readstream($f,4);
                                                    if($this-
>_readstream($f,4)!='IHDR')
                                                     $this-
>Error('Incorrect PNG file: '.$file);
                                                    $w =
$this->_readint($f);
                                                    $h =
$this->_readint($f);
                                                    $bpc =
ord($this->_readstream($f,1));
                                                    if($bpc>
8)
                                                     $this-
>Error('16-bit depth not supported: '.$file);
                                                    $ct =
ord($this->_readstream($f,1));
                                                    if($ct==0
|| $ct==4)

                                                    $colspac
e = 'DeviceGray';
                                                    elseif($ct
==2 || $ct==6)
```

```
                $colspace = 'DeviceRGB';
            elseif($ct==3)
                $colspace = 'Indexed';
            else
                $this->Error('Unknown color type: '.$file);
            if(ord($this->_readstream($f,1))!=0)
                $this->Error('Unknown compression method: '.$file);
            if(ord($this->_readstream($f,1))!=0)
                $this->Error('Unknown filter method: '.$file);
            if(ord($this->_readstream($f,1))!=0)
                $this->Error('Interlacing not supported: '.$file);
            $this->_readstream($f,4);
```

```
                                                                $dp =
'/Predictor 15 /Colors '.($colspace=='DeviceRGB' ? 3 : 1).'
/BitsPerComponent '.$bpc.' /Columns '.$w;


                                                                // Scan
chunks looking for palette, transparency and image data
                                                                $pal = '';
                                                                $trns =
'';
                                                                $data =
'';
                                                                do
                                                                {
                                                                 $n =
$this->_readint($f);
                                                                 $type =
$this->_readstream($f,4);

                                                                if($type=
='PLTE')
                                                                 {
                                                                        //
Read palette

                                                                $pal =
$this->_readstream($f,$n);
```

```
                                                    $this-
>_readstream($f,4);
                                                     }

                                                    elseif($ty
pe=='tRNS')
                                                     {
                                                            //
Read transparency info
                                                            $t
= $this->_readstream($f,$n);

                                                    if($ct==0
)

                                                    $trns =
array(ord(substr($t,1,1)));

                                                    elseif($ct
==2)

                                                    $trns =
array(ord(substr($t,1,1)), ord(substr($t,3,1)), ord(substr($t,5,1)));

                                                    else
                                                            {
```

```
                $pos =
strpos($t,chr(0));

                if($pos!=
=false)

                 $trns =
array($pos);

                    }

                $this-
>_readstream($f,4);

                 }

                elseif($ty
pe=='IDAT')

                 {

                    //
Read image data block

                $data .=
$this->_readstream($f,$n);

                $this-
>_readstream($f,4);

                 }
```

```
                                                    elseif($ty
pe=='IEND')

                                                    break;
                                                     else

                                                    $this-
>_readstream($f,$n+4);
                                                    }
                                                    while($n
);


                                                    if($colsp
ace=='Indexed' && empty($pal))
                                                     $this-
>Error('Missing palette in '.$file);
                                                    $info =
array('w'=>$w, 'h'=>$h, 'cs'=>$colspace, 'bpc'=>$bpc,
'f'=>'FlateDecode', 'dp'=>$dp, 'pal'=>$pal, 'trns'=>$trns);
                                                    if($ct>=4
)
                                                    {
                                                     //
Extract alpha channel
```

```
                                                    if(!functi
on_exists('gzuncompress'))

                                                    $this-
>Error('Zlib not available, can\'t handle alpha channel: '.$file);
                                                     $data =
gzuncompress($data);
                                                     $color
= '';
                                                     $alpha
= '';

                                                    if($ct==4
)
                                                     {
                                                          //
Gray image

                                                    $len =
2*$w;

                                                    for($i=0;
$i<$h;$i++)
                                                          {
```

```
                $pos =
(1+$len)*$i;

                $color .=
$data[$pos];

                $alpha
.= $data[$pos];

                $line =
substr($data,$pos+1,$len);

                $color .=
preg_replace('/(.)./s','$1',$line);

                $alpha
.= preg_replace('/.(.)/s','$1',$line);

                    }
                }
                else
                {
                    //
RGB image

                $len =
4*$w;
```

```
for($i=0;
$i<$h;$i++)
        {
$pos =
(1+$len)*$i;
$color .=
$data[$pos];
$alpha
.= $data[$pos];
$line =
substr($data,$pos+1,$len);
$color .=
preg_replace('/(.{3})./s','$1',$line);
$alpha
.= preg_replace('/.{3}(.)/s','$1',$line);
        }
 }
unset($d
ata);
```

```
                $data =
gzcompress($color);

                $info['s
mask'] = gzcompress($alpha);
                if($this-
>PDFVersion<'1.4')

                $this-
>PDFVersion = '1.4';
                }
                $info['da
ta'] = $data;
                return
$info;
}

function _readstream($f, $n)
{
                // Read n
bytes from stream
                $res = '';
                while($n
>0 && !feof($f))
                {
```

```
                                                    $s =
fread($f,$n);

                                                    if($s===f
alse)

                                                    $this-
>Error('Error while reading stream');

                                                    $n -=
strlen($s);

                                                    $res .=
$s;

                                                    }

                                                    if($n>0)

                                                    $this-
>Error('Unexpected end of stream');

                                                    return
$res;
}

function _readint($f)
{

                                                    // Read a
4-byte integer from stream

                                                    $a =
unpack('Ni',$this->_readstream($f,4));
```

```
                                                    return
$a['i'];
}

function _parsegif($file)
{
                                                    //
Extract info from a GIF file (via PNG conversion)
                                                    if(!functi
on_exists('imagepng'))
                                                     $this-
>Error('GD extension is required for GIF support');
                                                    if(!functi
on_exists('imagecreatefromgif'))
                                                     $this-
>Error('GD has no GIF read support');
                                                    $im =
imagecreatefromgif($file);
                                                    if(!$im)
                                                     $this-
>Error('Missing or incorrect image file: '.$file);
                                                    imageint
erlace($im,0);
                                                    $f =
@fopen('php://temp','rb+');
                                                    if($f)
```

```
                                                                {
                                                                 //
Perform conversion in memory
                                                                ob_start
();
                                                                imagepn
g($im);
                                                                 $data =
ob_get_clean();
                                                                imagede
stroy($im);
                                                                fwrite($f
,$data);
                                                                rewind($
f);
                                                                 $info =
$this->_parsepngstream($f,$file);
                                                                fclose($f
);
                                                                }
                                                                else
```

```
                                                                {
                                                                 // Use
temporary file
                                                                 $tmp =
tempnam('.','gif');

                                                                if(!$tmp)

                                                                $this-
>Error('Unable to create a temporary file');

                                                                if(!image
png($im,$tmp))

                                                                $this-
>Error('Error while saving to temporary file');

                                                                imagede
stroy($im);
                                                                 $info =
$this->_parsepng($tmp);

                                                                unlink($
tmp);
                                                                }
                                                                return
$info;
```

```
}

function _newobj()
{
                                                    // Begin
a new object
                                                    $this-
>n++;
                                                    $this-
>offsets[$this->n] = strlen($this->buffer);
                                                    $this-
>_out($this->n.' 0 obj');
}

function _putstream($s)
{
                                                    $this-
>_out('stream');
                                                    $this-
>_out($s);
                                                    $this-
>_out('endstream');
}

function _out($s)
```

```
{
                                                    // Add a
line to the document
                                                    if($this-
>state==2)
                                                     $this-
>pages[$this->page] .= $s."\n";
                                                    else
                                                     $this-
>buffer .= $s."\n";
}

function _putpages()
{
                                                    $nb =
$this->page;
                                                    if(!empt
y($this->AliasNbPages))
                                                    {
                                                     //
Replace number of pages
                                                    for($n=1
;$n<=$nb;$n++)
                                                    $this-
```

```
>pages[$n] = str_replace($this->AliasNbPages,$nb,$this-
>pages[$n]);
                                                    }
                                                    if($this-
>DefOrientation=='P')
                                                    {
                                                     $wPt =
$this->DefPageSize[0]*$this->k;
                                                     $hPt =
$this->DefPageSize[1]*$this->k;
                                                    }
                                                    else
                                                    {
                                                     $wPt =
$this->DefPageSize[1]*$this->k;
                                                     $hPt =
$this->DefPageSize[0]*$this->k;
                                                    }
                                                    $filter =
($this->compress) ? '/Filter /FlateDecode ' : '';
                                                    for($n=1
;$n<=$nb;$n++)
                                                    {
                                                     // Page
```

```
                $this->_newobj();
                $this->_out('<</Type /Page');
                $this->_out('/Parent 1 0 R');

                if(isset($this->PageSizes[$n]))
                $this->_out(sprintf('/MediaBox [0 0 %.2F %.2F]',$this->PageSizes[$n][0],$this->PageSizes[$n][1]));
                $this->_out('/Resources 2 0 R');

                if(isset($this->PageLinks[$n]))
                {
                        //Links
                $annots = '/Annots [';
                foreach($this->PageLinks[$n] as $pl)
```

```
                                                    {

                                                    $rect =
sprintf('%.2F %.2F %.2F
%.2F',$pl[0],$pl[1],$pl[0]+$pl[2],$pl[1]-$pl[3]);

                                                    $annots
.= '<</Type /Annot /Subtype /Link /Rect ['.$rect.'] /Border [0 0 0]
';

                                                    if(is_stri
ng($pl[4]))

                                                     $annots
.= '/A <</S /URI /URI '.$this->_textstring($pl[4]).'>>>>';

                                                    else

                                                    {

                                                     $l =
$this->links[$pl[4]];

                                                     $h =
isset($this->PageSizes[$l[0]]) ? $this->PageSizes[$l[0]][1] : $hPt;

                                                     $annots
```

```
.= sprintf('/Dest [%d 0 R /XYZ 0 %.2F null]>>',1+2*$l[0],$h-
$l[1]*$this->k);

                                                    }
                                                        }

                                                    $this-
>_out($annots.']');
                                                     }
                                                     if($this-
>PDFVersion>'1.3')

                                                    $this-
>_out('/Group <</Type /Group /S /Transparency /CS
/DeviceRGB>>');
                                                     $this-
>_out('/Contents '.($this->n+1).' 0 R>>');
                                                     $this-
>_out('endobj');
                                                     // Page
content
                                                     $p =
($this->compress) ? gzcompress($this->pages[$n]) : $this-
>pages[$n];
                                                     $this-
>_newobj();
```

```
                                                    $this-
>_out('<<'.$filter.'/Length '.strlen($p).'>>');
                                                    $this-
>_putstream($p);
                                                    $this-
>_out('endobj');
                                                   }
                                                   // Pages
root
                                                   $this-
>offsets[1] = strlen($this->buffer);
                                                   $this-
>_out('1 0 obj');
                                                   $this-
>_out('<</Type /Pages');
                                                   $kids =
'/Kids [';
                                                   for($i=0;
$i<$nb;$i++)
                                                    $kids .=
(3+2*$i).' 0 R ';
                                                   $this-
>_out($kids.']');
                                                   $this-
>_out('/Count '.$nb);
```

```
                                                    $this->_out(sprintf('/MediaBox [0 0 %.2F %.2F]',$wPt,$hPt));
                                                    $this->_out('>>');
                                                    $this->_out('endobj');
}

function _putfonts()
{
                                                    $nf = $this->n;
                                                    foreach($this->diffs as $diff)
                                                    {
                                                     //Encodings
                                                     $this->_newobj();
                                                     $this->_out('<</Type /Encoding /BaseEncoding /WinAnsiEncoding /Differences ['.$diff.']>>');
                                                     $this->_out('endobj');
                                                    }
```

```
                                                    foreach(
$this->FontFiles as $file=>$info)
                                                    {
                                                     // Font
file embedding
                                                     $this-
>_newobj();
                                                     $this-
>FontFiles[$file]['n'] = $this->n;
                                                     $font =
file_get_contents($this->fontpath.$file,true);

                                                    if(!$font)

                                                    $this-
>Error('Font file not found: '.$file);

                                                    $compre
ssed = (substr($file,-2)=='.z');

                                                    if(!$com
pressed && isset($info['length2']))

                                                    $font =
substr($font,6,$info['length1']).substr($font,6+$info['length1']+6,
$info['length2']);
```

```
                $this->_out('<</Length '.strlen($font));
                if($compressed)
                    $this->_out('/Filter /FlateDecode');
                $this->_out('/Length1 '.$info['length1']);
                if(isset($info['length2']))
                    $this->_out('/Length2 '.$info['length2'].' /Length3 0');
                $this->_out('>>');
                $this->_putstream($font);
                $this->_out('endobj');
            }
            foreach($this->fonts as $k=>$font)
            {
```

```
                                                    // Font
objects
                                                    $this-
>fonts[$k]['n'] = $this->n+1;
                                                    $type =
$font['type'];
                                                    $name
= $font['name'];

                                                   if($type=
='Core')
                                                    {
                                                          //
Core font

                                                   $this-
>_newobj();

                                                   $this-
>_out('<</Type /Font');

                                                   $this-
>_out('/BaseFont /'.$name);

                                                   $this-
>_out('/Subtype /Type1');
```

```
                                        if($name
!='Symbol' && $name!='ZapfDingbats')
                                        $this-
>_out('/Encoding /WinAnsiEncoding');
                                        $this-
>_out('>>');
                                        $this-
>_out('endobj');
                                         }
                                        elseif($ty
pe=='Type1' || $type=='TrueType')
                                         {
                                                //
Additional Type1 or TrueType/OpenType font
                                        $this-
>_newobj();
                                        $this-
>_out('<</Type /Font');
                                        $this-
>_out('/BaseFont /'.$name);
```

```
                $this->_out('/Subtype /'.$type);
                $this->_out('/FirstChar 32 /LastChar 255');
                $this->_out('/Widths '.($this->n+1).' 0 R');
                $this->_out('/FontDescriptor '.($this->n+2).' 0 R');
                if(isset($font['diffn']))
                    $this->_out('/Encoding '.($nf+$font['diffn']).' 0 R');
                else
                    $this->_out('/Encoding /WinAnsiEncoding');
                $this->_out('>>');
```

```
                $this->_out('endobj');
                //Widths
                $this->_newobj();
                $cw =&$font['cw'];
                $s = '[';
                for($i=32;$i<=255;$i++)
                $s .= $cw[chr($i)].' ';
                $this->_out($s.']');
                $this->_out('endobj');
                //Descriptor
```

```
                $this->_newobj();
                $s = '<</Type /FontDescriptor /FontName /'.$name;
                foreach($font['desc'] as $k=>$v)
                    $s .= ' /'.$k.' '.$v;
                if(!empty($font['file']))
                    $s .= ' /FontFile'.($type=='Type1' ? '' : '2').' '.$this->FontFiles[$font['file']]['n'].' 0 R';
                $this->_out($s.'>>');
                $this->_out('endobj');
            }
            else
            {
```

```
                                                                    //
Allow for additional types
                                                                    $mtd =
'_put'.strtolower($type);
                                                                    if(!meth
od_exists($this,$mtd))
                                                                    $this-
>Error('Unsupported font type: '.$type);
                                                                    $this-
>$mtd($font);
                                                                     }
                                                                    }
}

function _putimages()
{
                                                                    foreach(
array_keys($this->images) as $file)
                                                                    {
                                                                     $this-
>_putimage($this->images[$file]);
```

```
                unset($this->images[$file]['data']);
                unset($this->images[$file]['smask']);
            }
    }

    function _putimage(&$info)
    {
        $this->_newobj();
        $info['n'] = $this->n;
        $this->_out('<</Type /XObject');
        $this->_out('/Subtype /Image');
        $this->_out('/Width '.$info['w']);
        $this->_out('/Height '.$info['h']);
        if($info['cs']=='Indexed')
```

```
                $this->_out('/ColorSpace [/Indexed /DeviceRGB '.(strlen($info['pal'])/3-1).' '.($this->n+1).' 0 R]');
            else
            {
                $this->_out('/ColorSpace /'.$info['cs']);
                if($info['cs']=='DeviceCMYK')
                    $this->_out('/Decode [1 0 1 0 1 0 1 0]');
            }
            $this->_out('/BitsPerComponent '.$info['bpc']);
            if(isset($info['f']))
                $this->_out('/Filter /'.$info['f']);
            if(isset($info['dp']))
                $this->_out('/DecodeParms <<'.$info['dp'].'>>');
            if(isset($info['trns']) && is_array($info['trns']))
```

```
                                                                {
                                                                 $trns =
'';
                                                                for($i=0;
$i<count($info['trns']);$i++)
                                                                $trns .=
$info['trns'][$i].' '.$info['trns'][$i].' ';
                                                                 $this-
>_out('/Mask ['.$trns.']');
                                                                }
                                                                if(isset($
info['smask']))
                                                                 $this-
>_out('/SMask '.($this->n+1).' 0 R');
                                                                $this-
>_out('/Length '.strlen($info['data']).'>>');
                                                                $this-
>_putstream($info['data']);
                                                                $this-
>_out('endobj');
                                                                // Soft
mask
                                                                if(isset($
info['smask']))
```

```
                                                    {
                                                     $dp =
'/Predictor 15 /Colors 1 /BitsPerComponent 8 /Columns
'.$info['w'];
                                                     $smask
= array('w'=>$info['w'], 'h'=>$info['h'], 'cs'=>'DeviceGray',
'bpc'=>8, 'f'=>$info['f'], 'dp'=>$dp, 'data'=>$info['smask']);
                                                     $this-
>_putimage($smask);
                                                    }
                                                    // Palette
                                                    if($info['
cs']=='Indexed')
                                                    {
                                                     $filter =
($this->compress) ? '/Filter /FlateDecode ' : '';
                                                     $pal =
($this->compress) ? gzcompress($info['pal']) : $info['pal'];
                                                     $this-
>_newobj();
                                                     $this-
>_out('<<'.$filter.'/Length '.strlen($pal).'>>');
                                                     $this-
>_putstream($pal);
                                                     $this-
>_out('endobj');
```

```
                                                                    }
}

function _putxobjectdict()
{
                                                                    foreach(
$this->images as $image)
                                                                     $this-
>_out('/I'.$image['i'].' '.$image['n'].' 0 R');
}

function _putresourcedict()
{
                                                                    $this-
>_out('/ProcSet [/PDF /Text /ImageB /ImageC /ImageI]');
                                                                    $this-
>_out('/Font <<');
                                                                    foreach(
$this->fonts as $font)
                                                                     $this-
>_out('/F'.$font['i'].' '.$font['n'].' 0 R');
                                                                    $this-
>_out('>>');
                                                                    $this-
>_out('/XObject <<');
```

```
                                                    $this-
>_putxobjectdict();
                                                    $this-
>_out('>>');
}

function _putresources()
{
                                                    $this-
>_putfonts();
                                                    $this-
>_putimages();
                                                    //
Resource dictionary
                                                    $this-
>offsets[2] = strlen($this->buffer);
                                                    $this-
>_out('2 0 obj');
                                                    $this-
>_out('<<');
                                                    $this-
>_putresourcedict();
                                                    $this-
>_out('>>');
                                                    $this-
>_out('endobj');
```

```
}

function _putinfo()
{
                                                    $this->_out('/Producer '.$this->_textstring('FPDF '.FPDF_VERSION));
                                                    if(!empty($this->title))
                                                        $this->_out('/Title '.$this->_textstring($this->title));
                                                    if(!empty($this->subject))
                                                        $this->_out('/Subject '.$this->_textstring($this->subject));
                                                    if(!empty($this->author))
                                                        $this->_out('/Author '.$this->_textstring($this->author));
                                                    if(!empty($this->keywords))
                                                        $this->_out('/Keywords '.$this->_textstring($this->keywords));
                                                    if(!empty($this->creator))
```

```
                                                    $this->_out('/Creator '.$this->_textstring($this->creator));
                                                    $this->_out('/CreationDate '.$this->_textstring('D:'.@date('YmdHis')));
}

function _putcatalog()
{
                                                    $this->_out('/Type /Catalog');
                                                    $this->_out('/Pages 1 0 R');
                                                    if($this->ZoomMode=='fullpage')
                                                    $this->_out('/OpenAction [3 0 R /Fit]');
                                                    elseif($this->ZoomMode=='fullwidth')
                                                    $this->_out('/OpenAction [3 0 R /FitH null]');
                                                    elseif($this->ZoomMode=='real')
                                                    $this->_out('/OpenAction [3 0 R /XYZ null null 1]');
                                                    elseif(!is_string($this->ZoomMode))
```

```
                                                    $this->_out('/OpenAction [3 0 R /XYZ null null '.sprintf('%.2F',$this->ZoomMode/100).']');
                                                    if($this->LayoutMode=='single')
                                                    $this->_out('/PageLayout /SinglePage');
                                                    elseif($this->LayoutMode=='continuous')
                                                    $this->_out('/PageLayout /OneColumn');
                                                    elseif($this->LayoutMode=='two')
                                                    $this->_out('/PageLayout /TwoColumnLeft');
}

function _putheader()
{
                                                    $this->_out('%PDF-'.$this->PDFVersion);
}

function _puttrailer()
{
```

```
                                                    $this->_out('/Size '.($this->n+1));
                                                    $this->_out('/Root '.$this->n.' 0 R');
                                                    $this->_out('/Info '.($this->n-1).' 0 R');
}

function _enddoc()
{
                                                    $this->_putheader();
                                                    $this->_putpages();
                                                    $this->_putresources();
                                                    // Info
                                                    $this->_newobj();
                                                    $this->_out('<<');
                                                    $this->_putinfo();
                                                    $this->_out('>>');
```

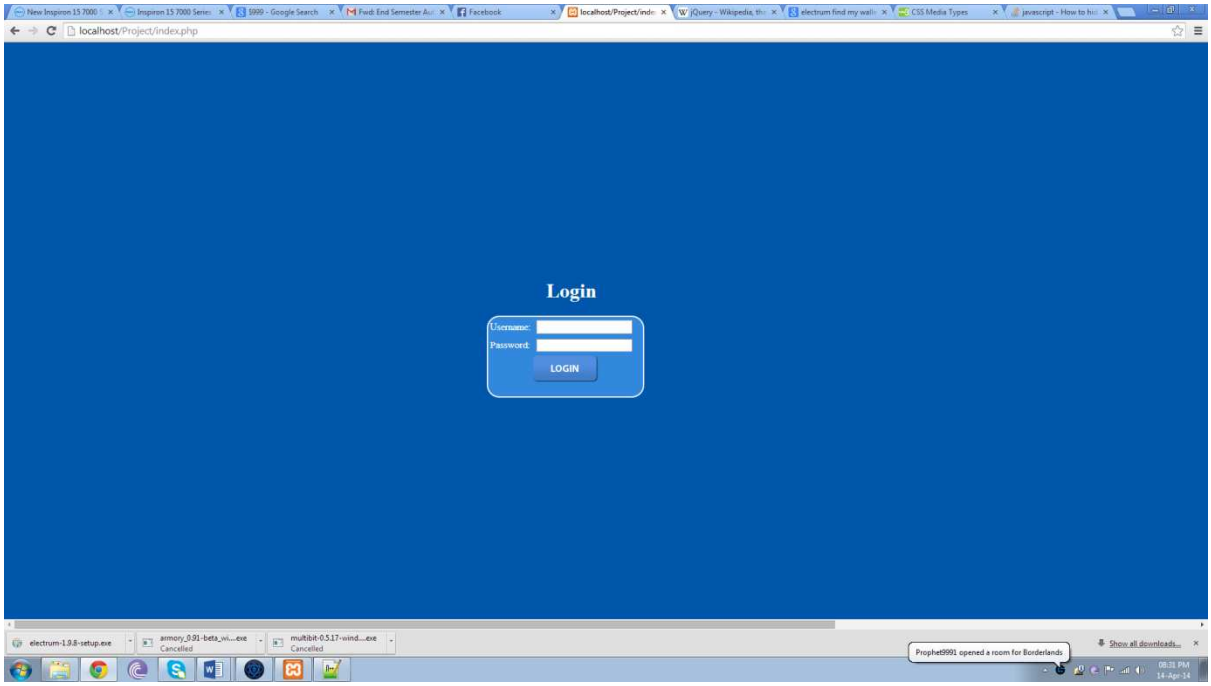
```
                                                    $this->_out('endobj');
                                                    // Catalog
                                                    $this->_newobj();
                                                    $this->_out('<<');
                                                    $this->_putcatalog();
                                                    $this->_out('>>');
                                                    $this->_out('endobj');
                                                    // Cross-ref
                                                    $o = strlen($this->buffer);
                                                    $this->_out('xref');
                                                    $this->_out('0 '.($this->n+1));
                                                    $this->_out('0000000000 65535 f ');
                                                    for($i=1; $i<=$this->n;$i++)
```

```
                                                    $this->_out(sprintf('%010d 00000 n ',$this->offsets[$i]));
                                                    // Trailer
                                                    $this->_out('trailer');
                                                    $this->_out('<<');
                                                    $this->_puttrailer();
                                                    $this->_out('>>');
                                                    $this->_out('startxref');
                                                    $this->_out($o);
                                                    $this->_out('%%EOF');
                                                    $this->state = 3;
}
// End of class
}

// Handle special IE contype request
if(isset($_SERVER['HTTP_USER_AGENT']) &&
$_SERVER['HTTP_USER_AGENT']=='contype')
```

```
{
                                                    header('
Content-Type: application/pdf');
                                                    exit;
}


?>
```

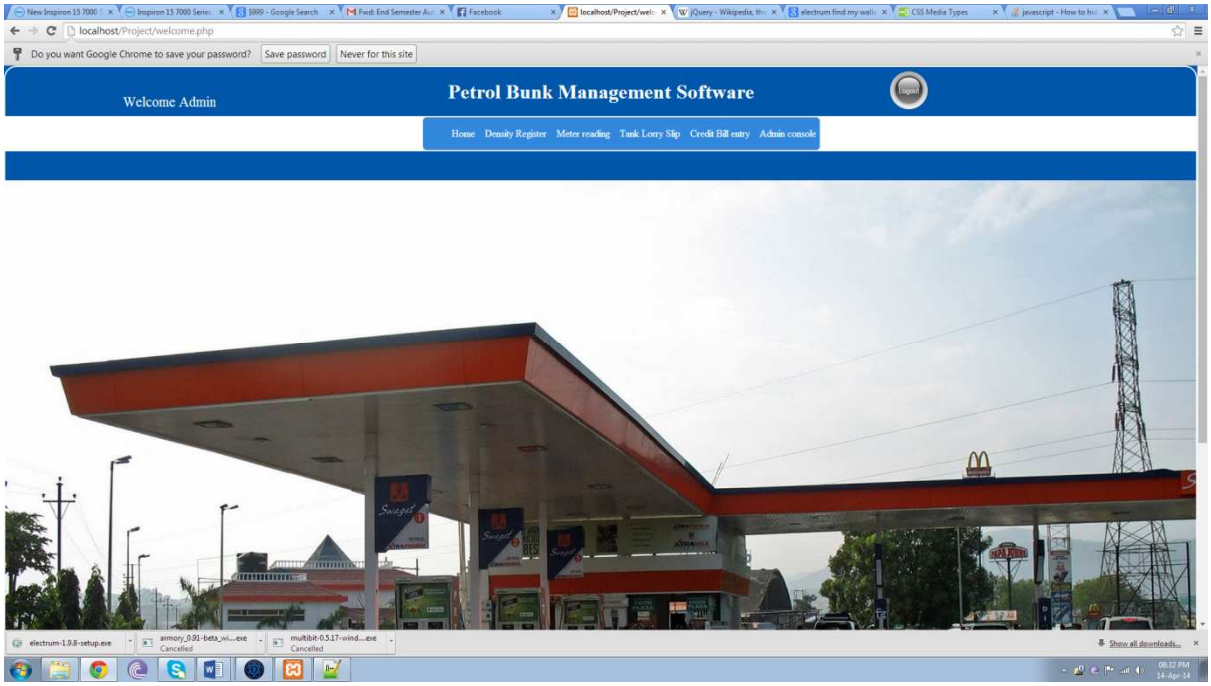
A.2 SCREEN SHOTS

Index/Login

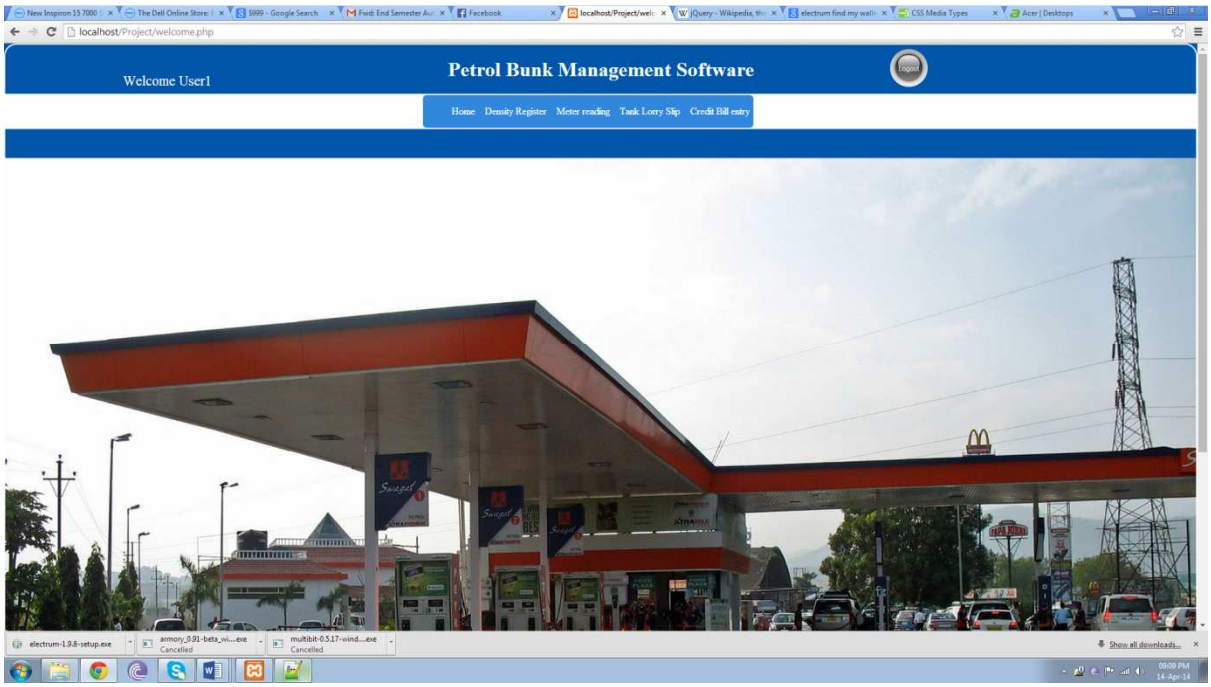


Welcome screen

Admin Login

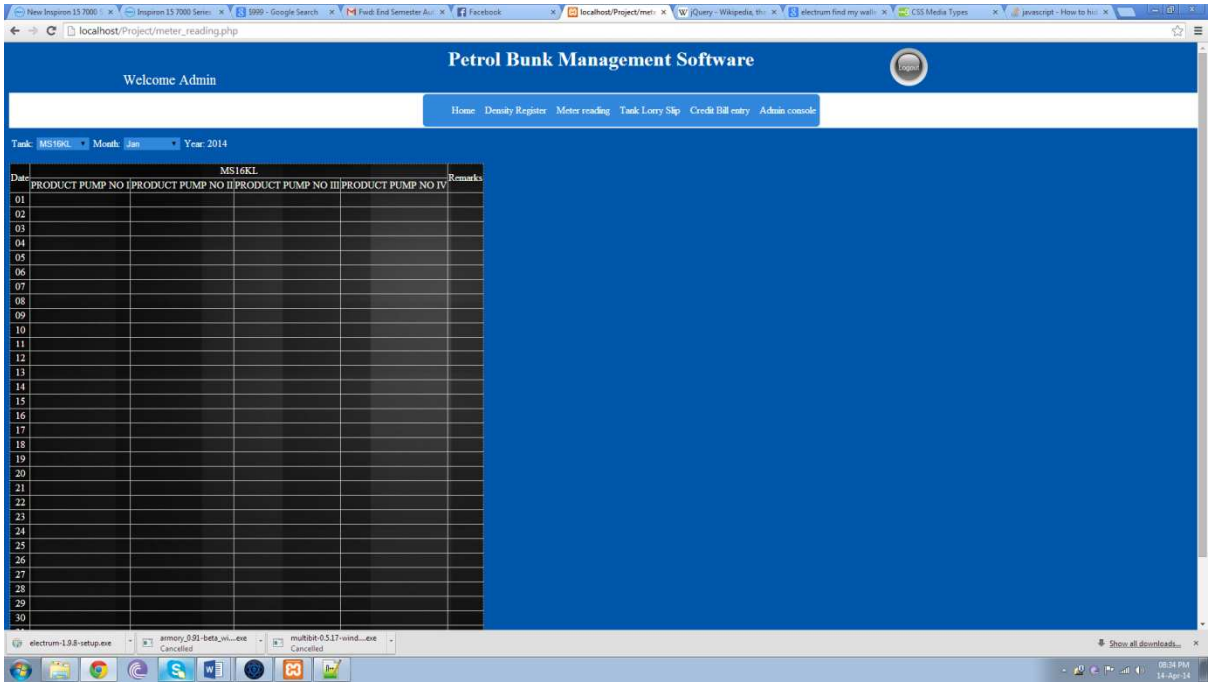


User login

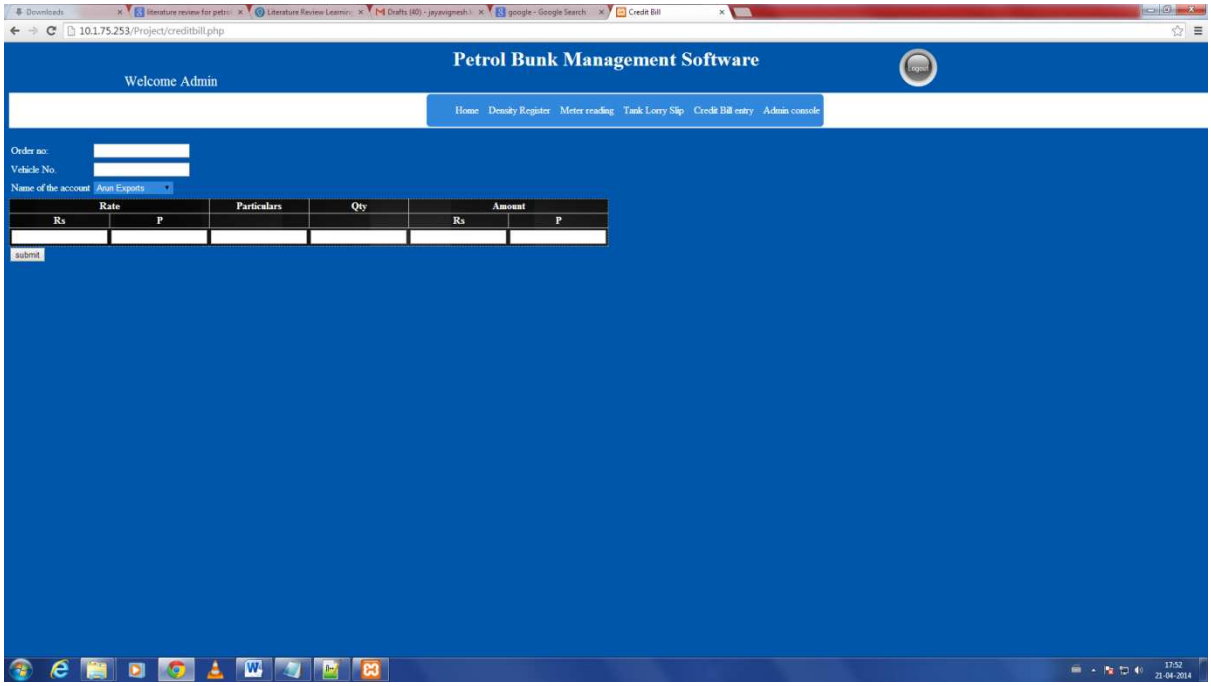


Density Register

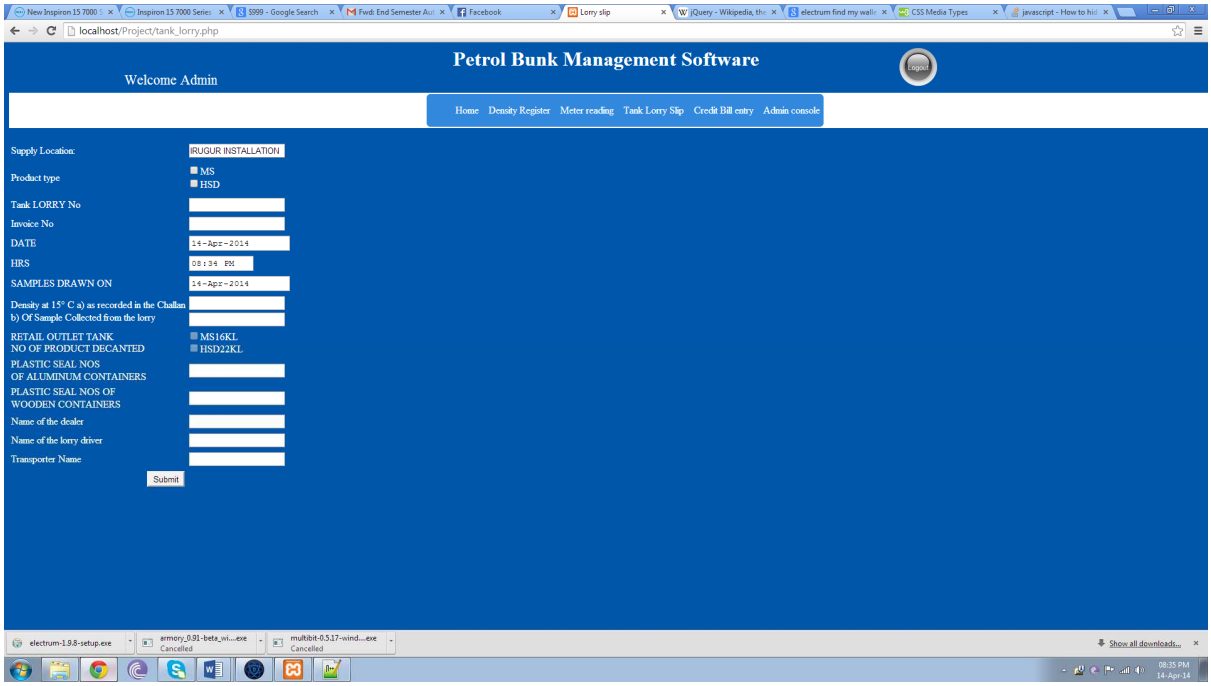
Meter reading



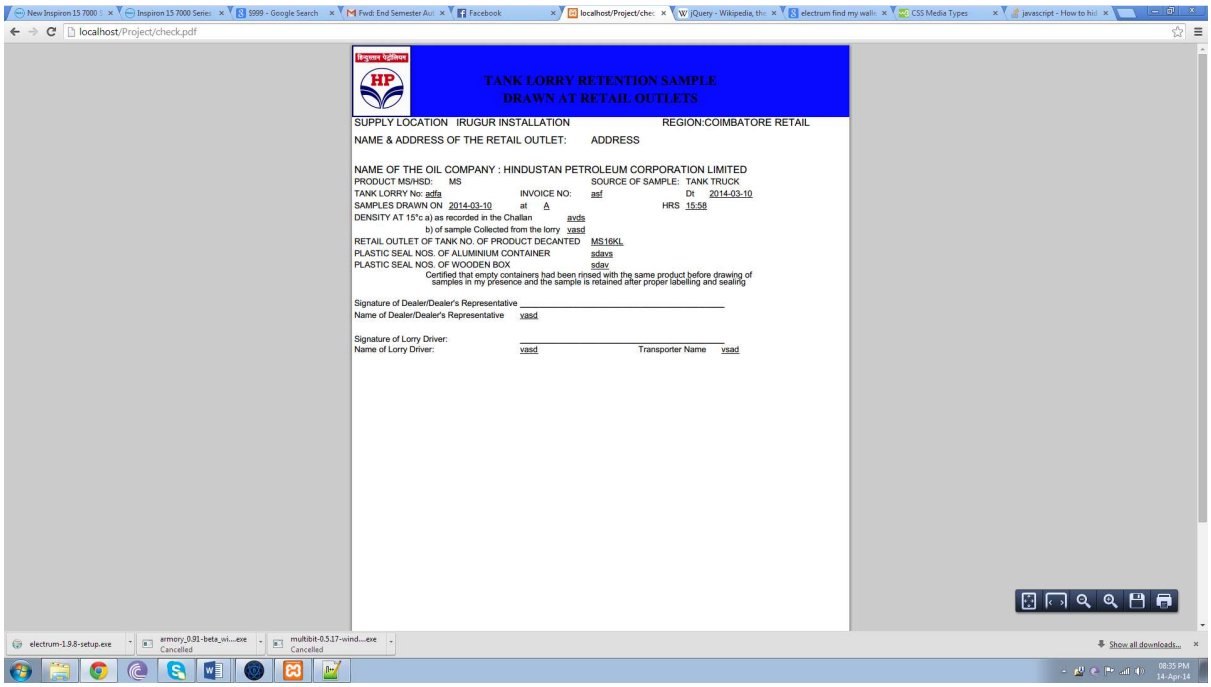
Credit bill party



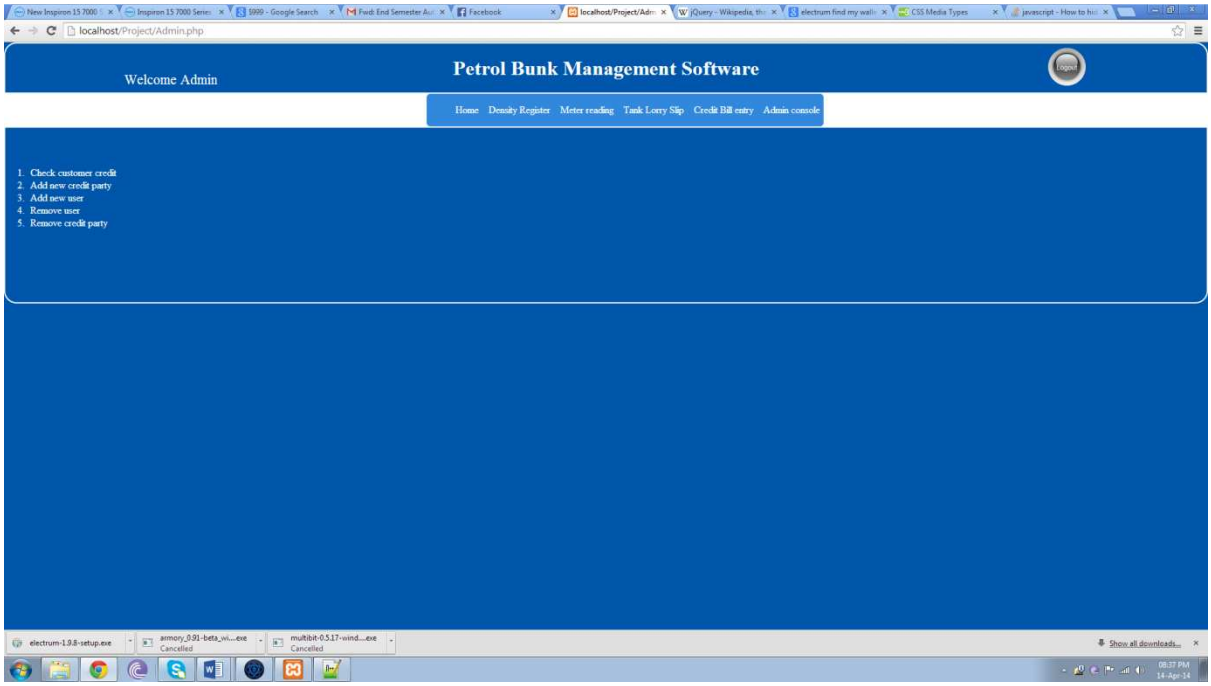
Tanker Lorry slip

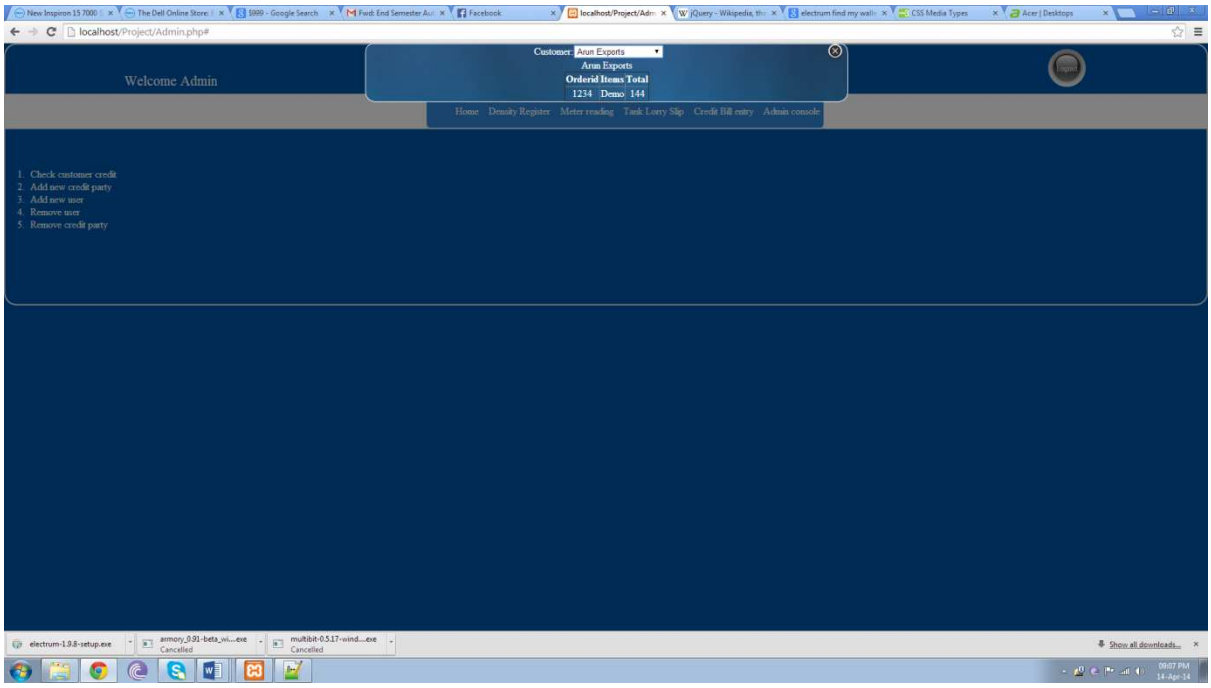


PDF



Admin console





REFERENCES

1. **W3 Schools-Online Tutorials, http://www.w3schools.com/**
2. **Density register calculator**