MONITORING OF PUBLIC TRANSPORTATION USING GPS

A PROJECT REPORT

Submitted by

PRADHEEP D **Reg No.: 13BEC105**

PRATHIBHA M **Reg No.: 13BEC106**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

ANNA UNIVERSITY : CHENNAI

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report **"MONITORING OF PUBLIC TRANSPORTATION USING GPS" is the bonafide work of PRADHEEP D [13BEC105] and PRATHIBHA M [13BEC106]** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Ms. A. Kalaiselvi M.E., (Ph.D),
PROJECT SUPERVISOR
Department of ECE
Kumaraguru College of Technology
Coimbatore-641049

SIGNATURE
Dr. K. Malarvizhi M.E., Ph.D,
HEAD OF THE DEPARTMENT
Department of ECE
Kumaraguru College of Technology
Coimbatore- 641049

The candidates with Register No: 13BEC105 and 13BEC106 are examined by us in the project viva-voce examination held on

……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere thanks to the Management of Kumaraguru College of Technology and Joint Correspondent **Shri. Shankar Vanavarayar** for the kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. R.S. Kumar, Ph.D.,** Kumaraguru College of Technology, who encouraged using each and every step of the project.

We would like to thank **Dr. K. Malarvizhi M.E., Ph.D.,** Head of the Department, Electronics and Communication Engineering, for her kind support and for providing necessary facilities to carry out the project work.

We wish to thank with everlasting gratitude to our Project Coordinator **Dr. A. Vasuki M.E., Ph.D.,** Department of Electronics and Communication Engineering for her consistent support throughout the course of this project work.

We are greatly privileged to express our deep sense of gratitude and heartfelt thanks to our Project Guide **Ms. A. Kalaiselvi, M.E., (Ph.D),** Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success and also we wish to convey our regards to all teaching and non-teaching staff of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support and abundant blessings in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

ABSTRACT

In today's time conscious world, each and every second is valuable. As Benjamin Franklin said, "*Lost time is never found again"*. Time once lost will not be regained. Keeping this in mind our project aims at providing a time chart of the availability of the buses at the bus stop. This helps the people to plan their journey according to the availability of the buses effectively, thus saving the time. In the project, we have developed an android app that sends the location of the bus continuously to the database at the APACHE server. The database also contains the location of the bus stops. With these datas, the time to reach each bus stop is calculated. Finally, by opening the webpage and giving the choice of the bus stop as the input, we can view the arrival time of the buses to that particular bus stop.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

IDE	Integrated Development Environment
ADT	Android Development Tool
SQL	Server Query Language
GPS	Global Positioning System
MVC	Model View Controller
URL	Universal Resource Locator
API	Application Programming Interface

CHAPTER 1

INTRODUCTION

Due to increasing traffic, public transportation has become unreliable to many users. Nowadays the number of personal vehicles like bikes and cars are increasing exponentially mainly because of inefficient public transportation. The main problem is that people waiting in the bus stop are not aware of the arrival of buses. One will have to wait for long hours awaiting the arrival of the desired bus. So, the project aims to solve this problem by making the people at the bus stop to be aware with arrival of buses.

The project aims at notifying the passengers about the time of the arrival of the buses. The passengers are given access through a webpage. The webpage has a drop-down box indicating the bus stops. The desired bus stop is chosen. This directs to a new page which displays the buses approaching the bus stop and the time at which they would arrive.

As a first step, the location of the bus travelling is obtained in the form of latitude and longitude coordinates with the help of GPS inbuilt in a phone. An android phone with marshmallow or higher versions are used. An android application is developed with Android studio. This android app first gets the busId as the input. Then it automatically fetches the coordinates of the bus. All the data is sent to the mentioned URL where they are stored for future processing.

The database is developed by inserting the data received from the android application. From the successive location of the bus, the speed at which the bus travels is determined. Further another table is created which lists down all the bus stops and the buses and their location coordinates. From the coordinates of the

buses and the bus stops the distance yet to be travelled to arrive at the bus stop is calculated. From the speed and distance obtained, the time that the bus would take to reach the bus stop is determined.

To provide access to the users to check the time chart, a webpage is created. In the webpage, there will be a drop-down box that displays the entire list of the bus stops among which the desired bus stop is chosen. This in turn refreshes the contents in the database and displays the busId, speed and time.

This project makes the public transportation more reliable so that the number of people using public transportation will increase. This will further ensure the minimal use of personal vehicles thereby reducing the burden on the environment and making it eco-friendly.

CHAPTER 2

PROJECT DESCRIPTION

The project involves transfer info of location of buses to the main server. This location is used to calculate the distance between the bus and the bus stops. The location of the bus stop is loaded in the server initially. By knowing the previous location and the current location of the bus we can calculate the speed of the bus. So, by dividing distance and speed we can obtain the time required for the bus to reach that particular bus stop.

2.1. BLOCK DIAGRAM

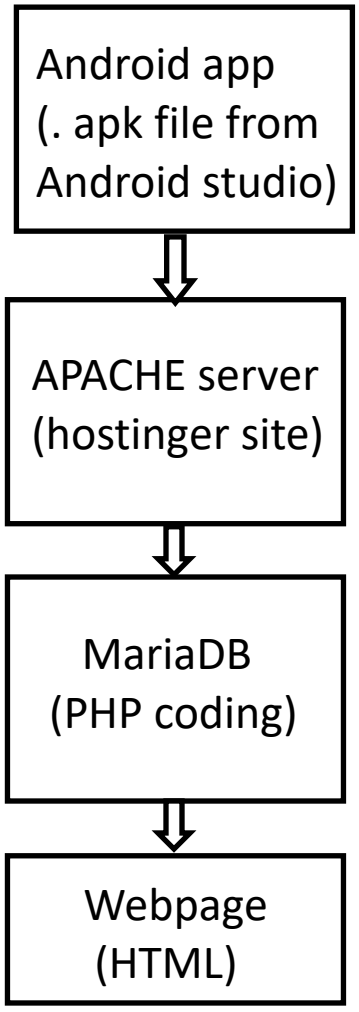


Fig:3.1-Block Diagram

2.2 ALGORITHM

Distance calculation

1. The coordinates of two locations for which the distance has to be calculated are taken (lat1,lon1) and (lat2,lon2)
2. temp1=sqrt{[(lat2-lat1)^2]+[(lon2-lon1)^2]} in terms of degree
3. To convert it into radians temp2=temp1*τ/180
4. Distance in kilometers is given by multiplying radius of earth, r=6371 km. ie. d=temp2*6371

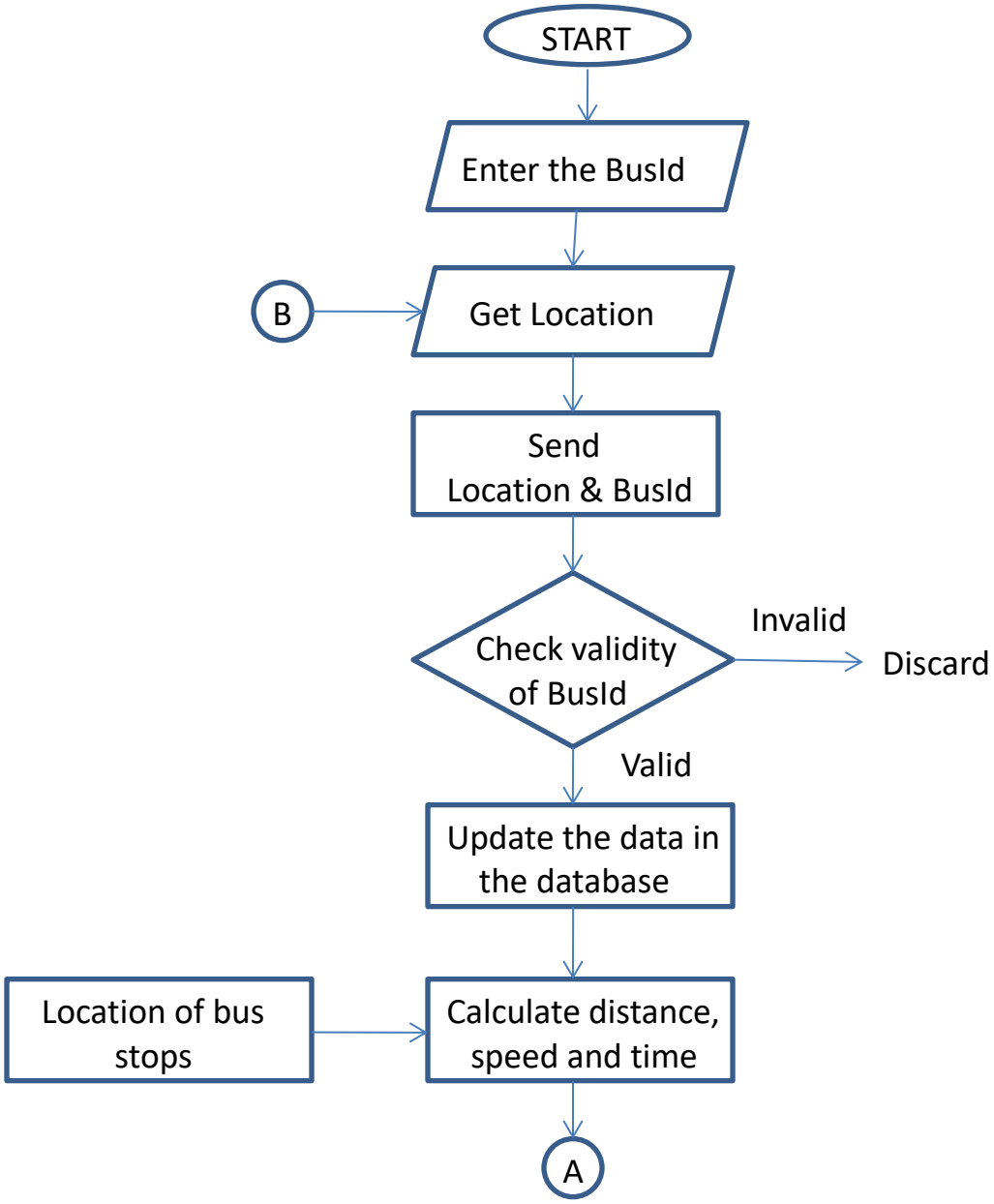
Speed calculation

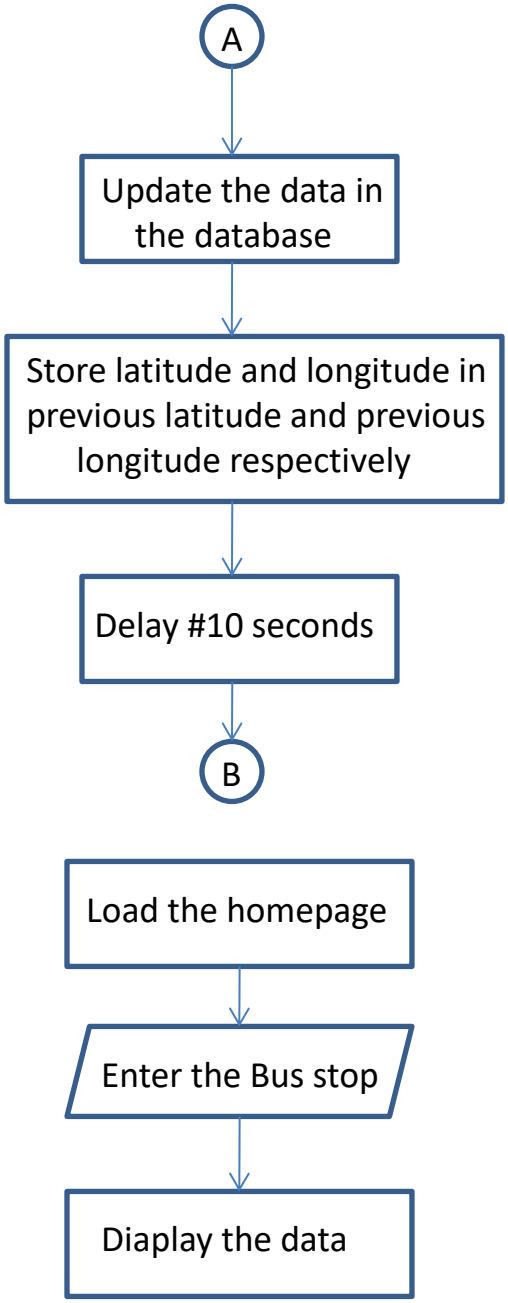
1. Calculate the distance, d (in meters) between two consecutive locations.
2. The time interval between two successive location is t. Here, t=10 seconds.
3. Speed = d/t (m/s)

Time calculation

1. Time is the ratio of distance to speed.
2. Time = distance/speed (seconds)

2.3 FLOW CHART




A
Update the data in the database
Store latitude and longitude in previous latitude and previous longitude respectively
Delay #10 seconds
B
Load the homepage
Enter the Bus stop
Diaplay the data

2.4. DESCRIPTION

The project has three main phases:

2.4.1 App Development for location sharing

2.4.2 Database creation

2.4.3 Webpage creation

2.4.1 APP DEVELOPMENT FOR LOCATION SHARING

Due to high accuracy and cost efficiency we have made use of GPS present in the smart phones in our project. In order to transfer the location from phone to the server, we have created an application. The application has been named as *Bus Info*. We have made use of Android Studio to develop this application. Android Studio is the official Integrated Development Environment (IDE) for the Android platform. Android Studio is designed specifically for Android development. It is available for download on Windows, macOS and Linux, and replaced Eclipse Android Deveopment Tools(ADT) as Google's primary IDE for native Android application development.

The following features are available in the current stable version of Android Studio

- Gradle based build support.
- Android-specific refactoring and quick fixes.
- Lint tools to catch performance, usability, version compatibility and other problems.
- ProGuard integration and app-signing capabilities.
- Template-based wizards to create common Android designs and components.
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.

- Support for building Android Wear apps.
- Built-in support for Google Cloud Platform, enabling integration with Google Cloud Messaging and App Engine.
- An Android Virtual Device that is used to run and debug apps.

2.4.2 DATABASE CREATION

In order to establish a centralized monitoring over the public transportation we have created a database. The information regarding buses and bus stops are stored in the database. We have created our database in the open source platform called *Hostinger*. Numerous web hosting sevices are available today. But Hostinger seems to be a better choice for hosting because it is reliable and free of cost. Hostinger was bootstrapped in 2004. The idea of Hostinger is to create a website for free, with no limits PHP, MySQL, cPanel. It has 29 million users in 178 countries. 15K new sign-ups on average every day - that's 1 new client every 5 seconds. Hostinger provides options of creating database with support of MariaDB. It also allows us to build our own sites with various Import and Export options. Its bandwidth is about 100GB.

2.4.2.1 MariaDB

The Hostinger website uses MariaDB. MariaDB is a software programming language similar to that of MySQL.

SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). SQL server is owned by Microsoft and is typically

referred as **Microsoft SQL Server**. It has a long history of releases and it is updated often adding all latest trends and technologies to it thus making it one of the trusted data base applications today. But it is not open source.

MySQL

The MySQL development project is owned by Oracle corporation has made its source code available under the terms of the GNU General Public License. So it can be accessed by everyone without any license.

MariaDB is a community-developed software of the MySQL relational database management system intended to remain free for the users. MariaDB intends to maintain high compatibility with MySQL, ensuring a "drop-in" replacement capability with library binary equivalency and exact matching with MySQL APIs and commands. It includes the XtraDB storage engine for replacing InnoDB as well as a new storage engine Aria, that intends to be both a transactional and non-transactional engine perhaps even included in future versions of MySQL

MariaDB is also an open source software developed on the basis of MySQL, but MariaDB has following advantages:

- Quicker and more transparent security releases
- MariaDB development is more open and vibrant
- More storage engines
- Galera active-active master clustering

2.4.2.2 XAMPP

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well. It is a simple, lightweight Apache distribution that makes it extremely easy to create a local web server for testing and deployment purposes. XAMPP's is intended to be used as development tool, which enables us to test the work on computers without any access to the Internet. XAMPP is a open source cross platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Once XAMPP is installed, it is possible to treat a local host like a remote host by connecting using an FTP client. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable filed.

2.4.3 WEBPAGE CREATION

In order to display the data i.e the timings of the buses to the users we have created a webpage. So with the help of webpage we can view the arrival of buses anywhere within a fraction of second. The webpage is also displayed in the bus stops so that people in the bus stops can know the arrival of buses. We have used *CodeIgniter* and *Materialize* to design the webpage. Materialize was created and designed by Google, Material Design is a design language that combines the classic principles of successful design along with innovation and technology.

2.4.3.1 CodeIgniter

CodeIgniter is an open source software rapid development web framework, for use in building dynamic web sites with PHP. CodeIgniter is most often noted *"because it is faster, lighter and the least like a framework".* The first public version of CodeIgniter was released by EllisLab on February 28, 2006. CodeIgniter's source code is maintained at GitHub, and it is the certified open source software licensed with the MIT License.

Some of the principal benefits in CodeIgniter are:

- It makes coding in PHP simple, quick and user-friendly.
- It's an excellent framework for learning more about how PHP works as we code.
- It underpins the Model/View/Controller (MVC) approach to web development.
- It's built on a linear, easy-to-use folder structure.
- It's open source and simple to configure and customize.
- Small footprint with exceptional performance
- MVC approach to development (although it is very loosely based which allows for flexibility)
- Generates search engine friendly clean URLs
- Easily extensible
- Runs on both PHP 4 (4.3.2+) and 5
- Application security is a focus
- Easy caching operations
- Most libraries are only loaded when needed which cuts back on resources needed.

CodeIgniter is loosely based on the popular model–view–controller (MVC) development pattern.

MVC stands for Model, View, Controller. It is a programming pattern used in developing web apps. This pattern isolates the user interface and backend. MVC also increases the flexibility of an app by being able to reuse models or views over again.

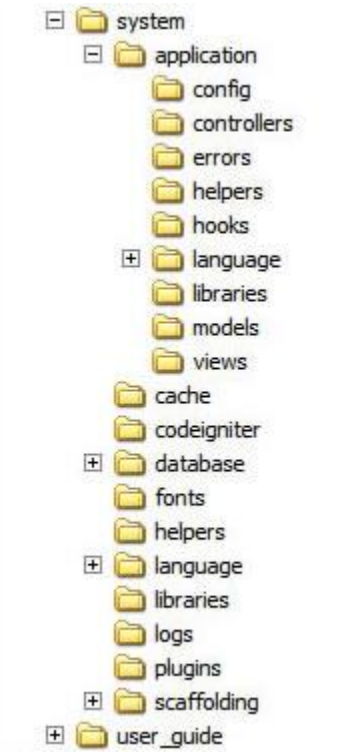
Below is a description of MVC.

- **Model:** The model deals with the raw data and database interaction and will contain functions like adding records to a database or selecting specific database records. In CI the model component is not required and can be included in the controller.
- **View:** The view deals with displaying the data and interface controls to the user with. In CI the view could be a web page, rss feed, ajax data or any other "page".
- **Controller:** The controller acts as the in between of view and model and, as the name suggests, it controls what is sent to the view from the model. In CI, the controller is also the place to load libraries and helpers.

An example of a MVC approach can be explained as:

1. The user interacts with the view by filling in a form and submitting it.
2. The controller receives the POST data from the form, the controller sends this data to the model which updates in the database.
3. The model then sends the result of the database to the controller.
4. This result is updated in the view and displayed to the user.

The basic structure of CodeIgniter is as follows:



- The **system** folder stores all the files which make CI work.
    - The **application** folder is almost identical to the contents of the **system** folder this is so the user can have files that are particular to that application.
        - The **config** folder stores all the config files relevant to the application. Which includes information on what libaries the application should auto load and database details.
        - The **controllers** folder stores all the controllers for the application.
        - The **errors** folder stores all the template error pages for the application. When an error occurs the error page is generated from one of these templates.
        - The **helpers** folder stores all the helpers which are specific to your application.

    - The **hooks** folder is for hooks which modify the functioning of CI's core files, hooks should only be used by advanced users of CI
    - The **language** folder stores lines of text which can be loaded through the language library to create multilingual sites.
    - The **libraries** folder stores all the libraries which are specific to your application.
    - The **models** folder stores all the models for the application.
    - The **views** folder stores all the views for the application.
- The **cache** folder stores all the caches generated by the caching library.
- The **codeigniter** folder stores all the internals which make CI work.
- The **database** folder stores all the database drivers and class which enable you to connect to database.
- The **fonts** folder stores all the fonts which can be used by the image manipulation library.
- The **helpers** folder stores all of CI's core helpers but you can place your own helpers in here which can be accessed by all of your applications.
- The **language** folder stores all of CI's core language files which its libaries and helpers use. You can also put your own language folders which can accessed by all of your applications.
- The **libaries** folder stores all of CI's core libaries but you can place your own libraries in here which can be accessed by all of your applications
- The **logs** folder stores all of the logs generated by CI.
- The **plugin** folder stores all of the plugins which you can use. Plugins are almost identical to helpers, plugins are functions intended to be shared by the community.

- The **scaffolding** folder stores all the files which make the scaffolding class work. Scaffolding provides a convenient CRUD like interface to access information in your database during development.

- The **user guide** houses the user guide to CI.
- The **index.php** file is the bit that does all the CI magic it also lets the you change the name of the **system** and **application** folders.

2.4.3.2 SUBLIME TEXT

The PHP code is written in a software called Sublime Text. Sublime Text supports various other languages like JavaScript, Perl, Python, Ruby. Sublime Text is a proprietary cross platform source code editor with a Python Application Programming Interface (API). It natively supports many programming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free software licenses.

The following is a list of features of Sublime Text:

- "Goto Anything," quick navigation to files, symbols, or lines.
- "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands.
- Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas.
- Python-based plugin API.
- Project-specific preferences.
- Extensive customizability via JSON settings files, including project-specific and platform-specific settings.
- Cross platform (Windows, OS X, Linux)

- Compatible with many language grammars from TextMate.
- Auto-save, which attempts to prevent users from losing their work
- Customizable key bindings, a navigational tool which allows users to assign hotkeys to their choice of options in both the menus and the toolbar.
- Find as we type, begins to look for the text being entered as the user types without requiring a separate dialog box.
- Spell check function corrects as we type.
- Macros.
- Repeat the last action.

CHAPTER 3

RESULTS

This project provides a time chart that helps the commuters to plan their travel effectively. An important feature incorporated in this is that anyone can view the time chart from anywhere accessing the webpage.

In the fig. 3.1 we can see the android application developed for transmitting the data. This application first receives the Bus ID as the input. Then it gets the coordinates through GPS followed by transmission of the data to the mentioned URL. This continues for every 10 seconds with the help of a timer.

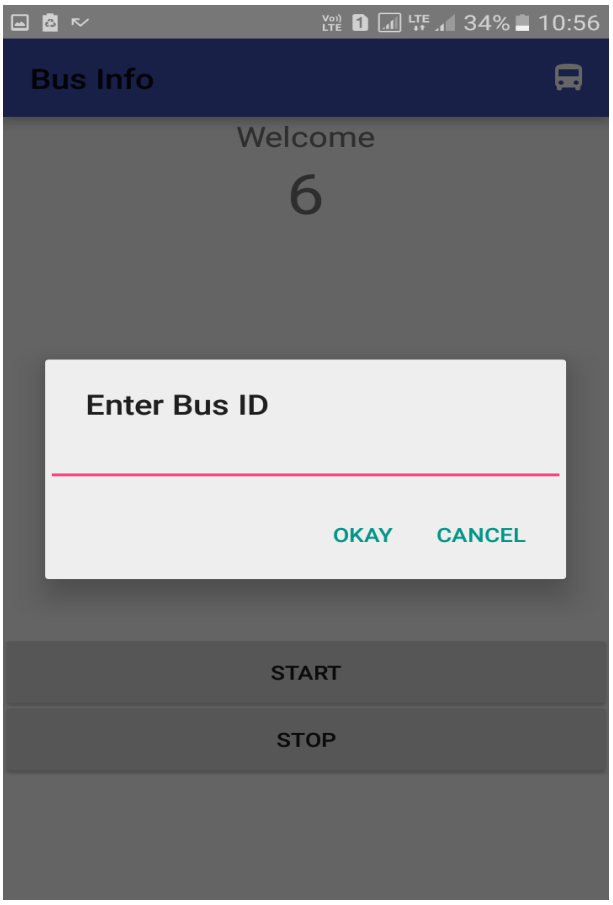


Fig:3.1-App for location sharing

Fig. 3.2 displays the table with the location updates of the buses in the MariaDB database. The table has a set of predefined busId's. When the data from the application is received, the busId is first checked and then the row with the same busId is updated with the received latitude and longitude values. Further the previous latitude and previous longitude are also updated to calculate the speed with which the bus is travelling.

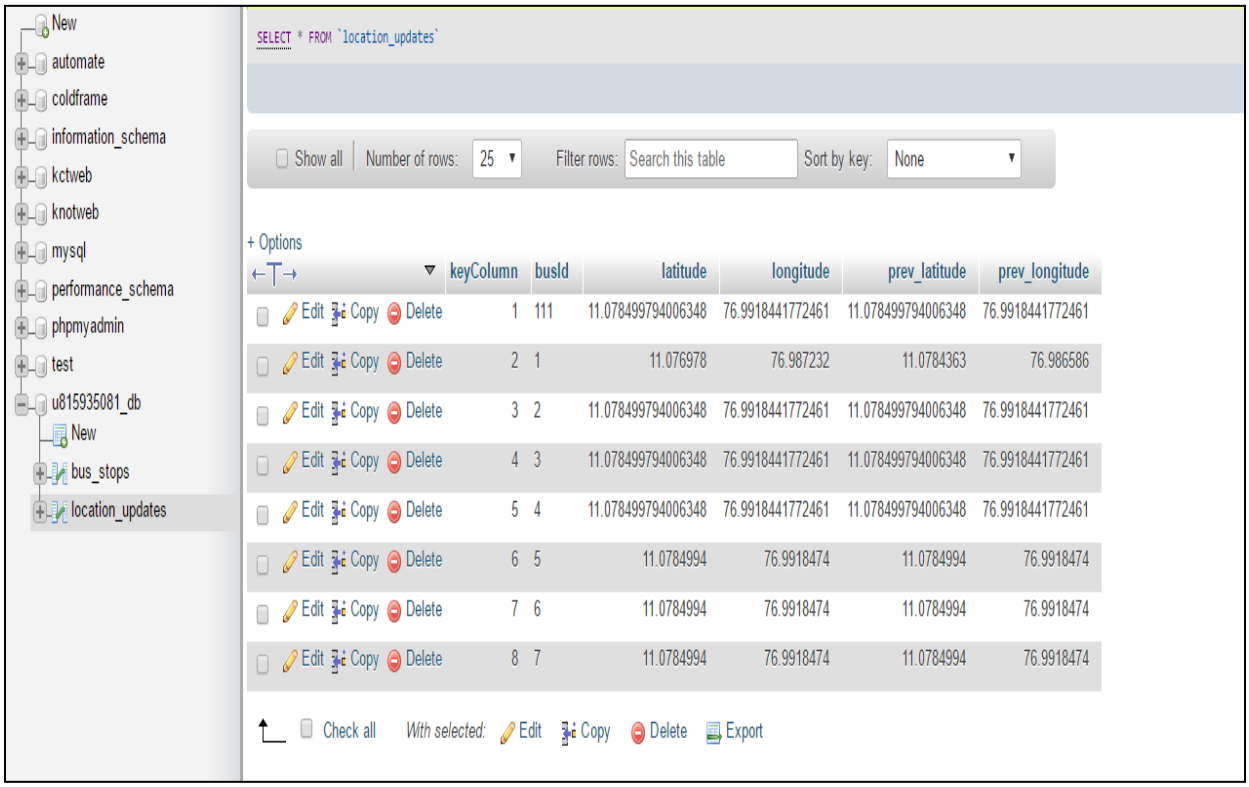
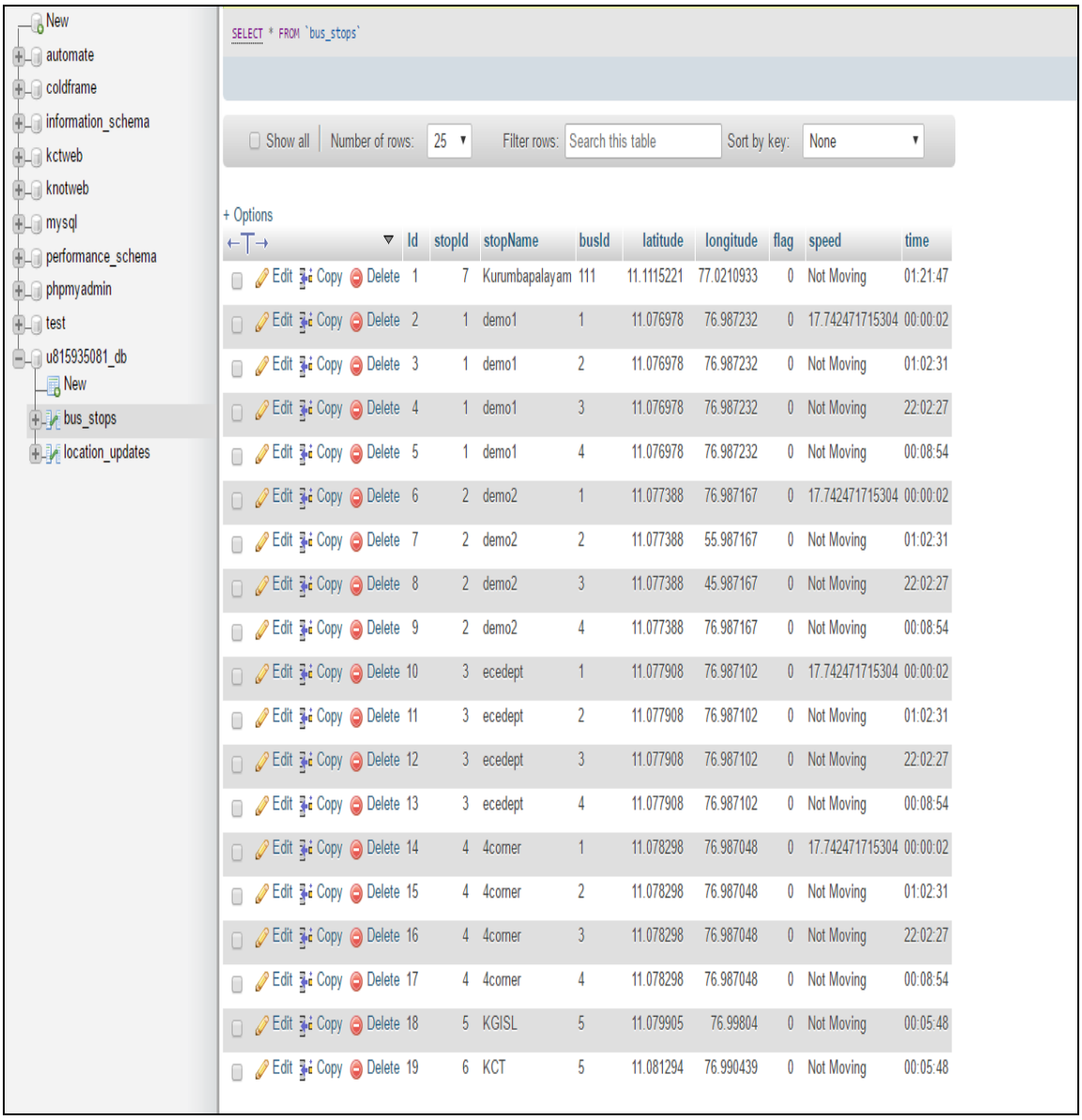


Fig:3.2-Database of location of buses

Fig. 3.3 shows the list of bus stops with the buses that would pass through that particular bus stop. When a particular bus stop is chosen in the webpage, the distance, speed and time of all the buses approaching the bus stop are calculated from the coordinates of the bus stop available in this table and the coordinates of the location of buses available in the location updates table and are updated in this table.



	Id	stopId	stopName	busId	latitude	longitude	flag	speed	time
Edit Copy Delete	1	7	Kurumbapalayam	111	11.1115221	77.0210933	0	Not Moving	01:21:47
Edit Copy Delete	2	1	demo1	1	11.076978	76.987232	0	17.742471715304	00:00:02
Edit Copy Delete	3	1	demo1	2	11.076978	76.987232	0	Not Moving	01:02:31
Edit Copy Delete	4	1	demo1	3	11.076978	76.987232	0	Not Moving	22:02:27
Edit Copy Delete	5	1	demo1	4	11.076978	76.987232	0	Not Moving	00:08:54
Edit Copy Delete	6	2	demo2	1	11.077388	76.987167	0	17.742471715304	00:00:02
Edit Copy Delete	7	2	demo2	2	11.077388	55.987167	0	Not Moving	01:02:31
Edit Copy Delete	8	2	demo2	3	11.077388	45.987167	0	Not Moving	22:02:27
Edit Copy Delete	9	2	demo2	4	11.077388	76.987167	0	Not Moving	00:08:54
Edit Copy Delete	10	3	ecedept	1	11.077908	76.987102	0	17.742471715304	00:00:02
Edit Copy Delete	11	3	ecedept	2	11.077908	76.987102	0	Not Moving	01:02:31
Edit Copy Delete	12	3	ecedept	3	11.077908	76.987102	0	Not Moving	22:02:27
Edit Copy Delete	13	3	ecedept	4	11.077908	76.987102	0	Not Moving	00:08:54
Edit Copy Delete	14	4	4corner	1	11.078298	76.987048	0	17.742471715304	00:00:02
Edit Copy Delete	15	4	4corner	2	11.078298	76.987048	0	Not Moving	01:02:31
Edit Copy Delete	16	4	4corner	3	11.078298	76.987048	0	Not Moving	22:02:27
Edit Copy Delete	17	4	4corner	4	11.078298	76.987048	0	Not Moving	00:08:54
Edit Copy Delete	18	5	KGISL	5	11.079905	76.99804	0	Not Moving	00:05:48
Edit Copy Delete	19	6	KCT	5	11.081294	76.990439	0	Not Moving	00:05:48

Fig:3.3-Database of bus stops

Fig. 3.4 shows the homepage. The homepage displays the name of the page along with a search and refresh tab. It also has a drop down box in which the choice of the bus stop has to chosen out of the predefined names. By clicking on the submit button after choosing the bus stop, it gets navigated to the corresponding page of the bus stop.

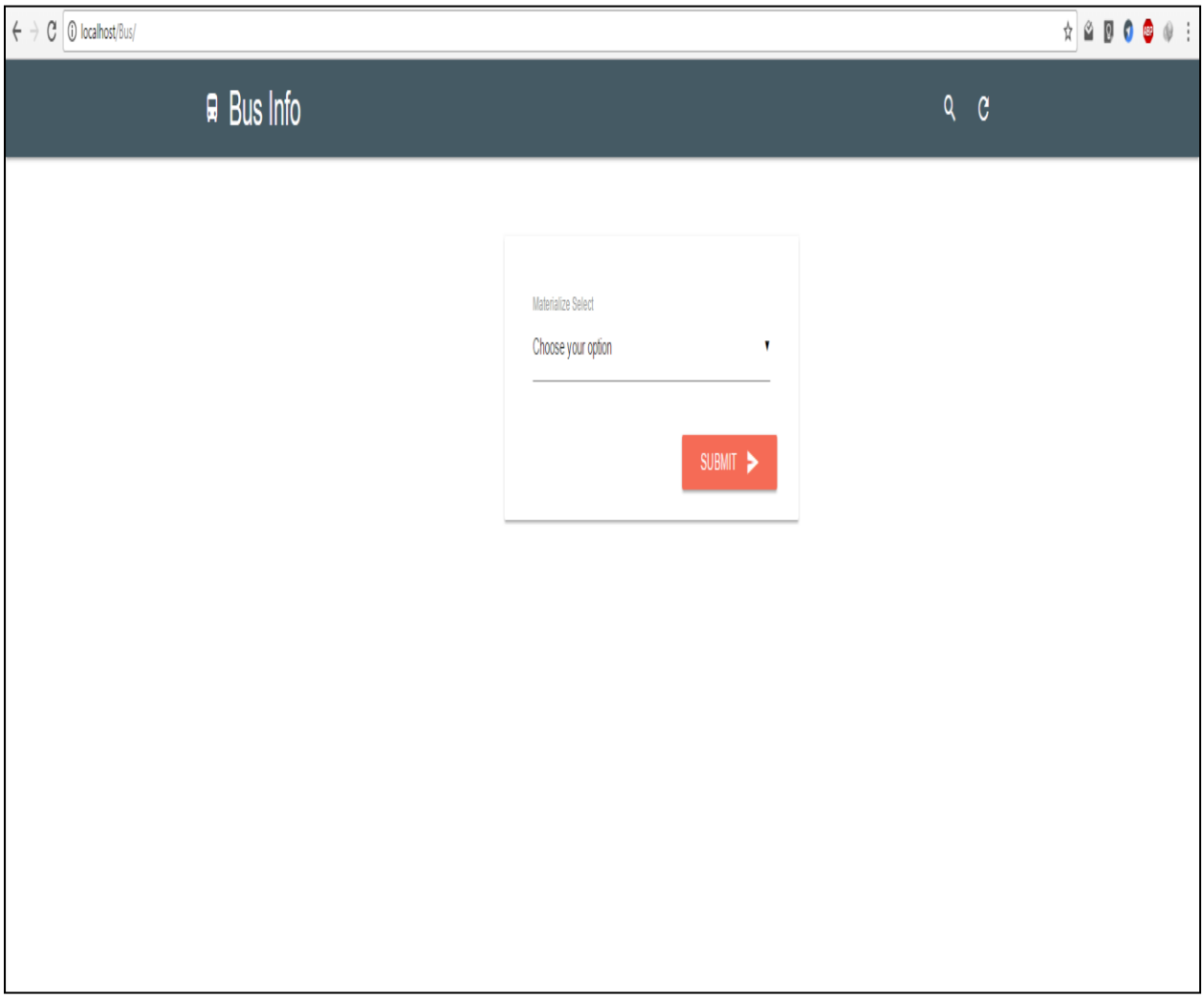


Fig:3.4-Homepage

In Fig. 3.5 the redirected page from the homepage is shown. When a bus stop is chosen, and submitted in the homepage, a new page listing the buses approaching the specified bus stop is displayed along with the speed with which the bus is travelling and the time that the bus would take to arrive at the bus stop.

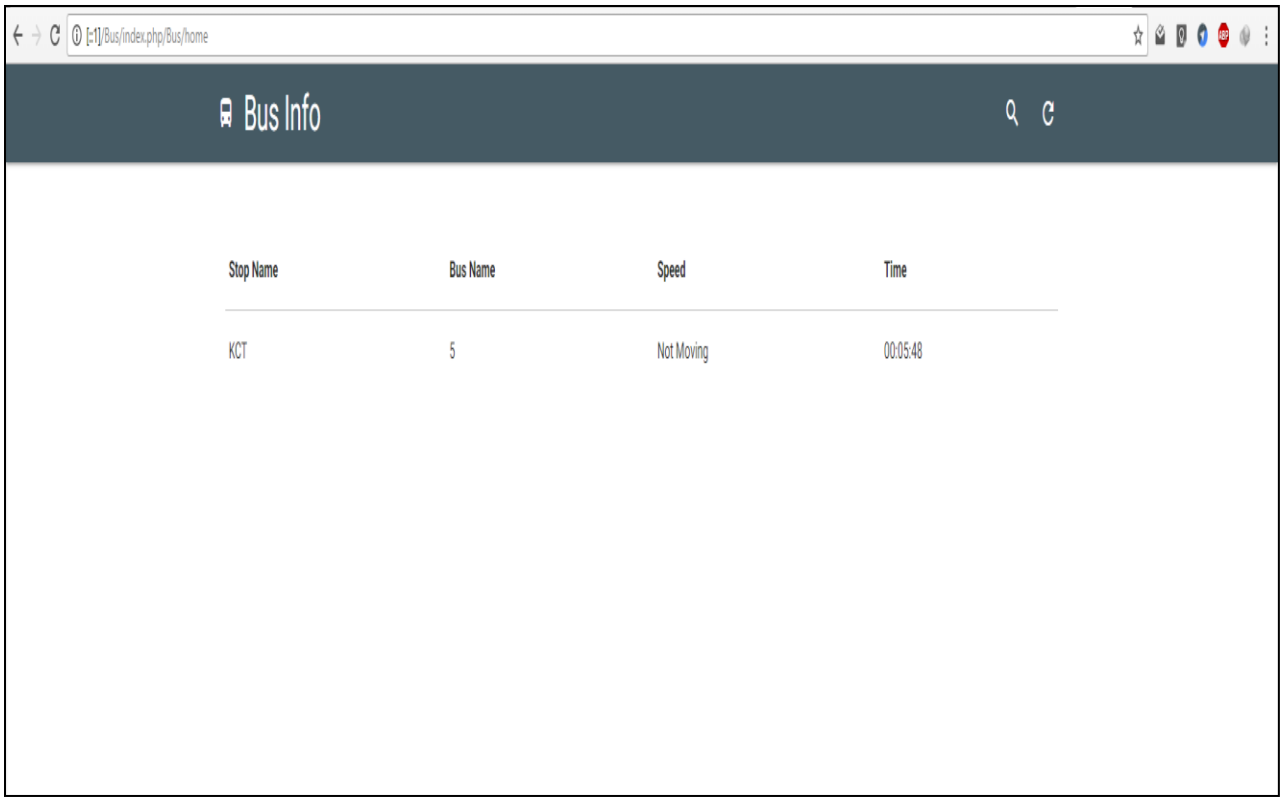


Fig:3.5-Time chart of buses

CHAPTER 4

CONCLUSION

"If time be of all things the most precious, wasting time must be the greatest prodigality". So, this project is a "One step solution for smart and effective transportation" which makes the public transportation more efficient. With the aid of this project the passengers will be able to plan their travel more wisely. There is only a minimum probability of setback in the schedule devised. This helps the commuters in saving their precious time that they would normally spend waiting at the bus stops. Since the use of personal transportation requires loss of energy, people would opt for public transportation which is timely rather than the earlier ways. So, this solution will encourage people to use public transportation. This in turn minimizes the traffic. Also, due to the use of public transportation the fuel is conserved along with being eco- friendly due to reduced levels of pollution.

DRAWBACKS

The drawback of the project is that this can be devised in areas only with network accessibility. This cannot be employed in rural areas were this would play a predominant role due to the lack of network facilities. Also, the distance algorithm is devised keeping straight roads in mind. This would prove less effective if irregular roads are found. In addition to this, in an irregular terrain like mountains, the altitude factor would cause an uncertainty in the distance calculation thus indirectly affecting the time.

FUTURE SCOPE

The further enhancement that can be made is that we can fetch the data from the google maps to know the routes thereby making the time calculations more accurate. We can also include the information regarding live traffic to make the monitoring more efficient.

REFERENCE

1. https://www.youtube.com/watch?v=lvcGh2ZgHeA
2. https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all
3. https://www.w3schools.com/php/php_syntax.asp
4. http://materializecss.com/getting-started.html
5. https://codeigniter.com/

APPENDICES

APPENDIX A

Sample code written in Android Studio for android application development

```
public Location getLocation() {

try {

locationManager = (LocationManager) mContext.getSystemService
(LOCATION_SERVICE);

// getting GPS status

isGPSEnabled = locationManager.isProviderEnabled
(LocationManager.GPS_PROVIDER);

// getting network status

isNetworkEnabled = locationManager.isProviderEnabled
(LocationManager.NETWORK_PROVIDER);

if (!isGPSEnabled &amp;amp;&amp;amp; !isNetworkEnabled) {

// no network provider is enabled

} else {

this.canGetLocation = true;

if (isNetworkEnabled) {

locationManager.requestLocationUpdates(LocationManager.NETWORK_PRO
VIDER, MIN_TIME_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATE ,  this);

Log.d(&amp;quot;Network&amp;quot;, &amp;quot;Network&amp;quot;);
```

```
if (locationManager != null) {

location = locationManager.getLastKnownLocation
(LocationManager.NETWORK_PROVIDER);

if (location != null) {

latitude = location.getLatitude();

longitude = location.getLongitude();

}

}

}

// if GPS Enabled get lat/long using GPS Services

if (isGPSEnabled) {

if (location == null) {

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
MIN_TIME_BW_UPDATES,MIN_DISTANCE_CHANGE_FOR_UPDATES,
this);

Log.d(&amp;quot;GPS Enabled&amp;quot;, &amp;quot;GPS Enabled&amp;quot;);

if (locationManager != null) {

location = locationManager.getLastKnownLocation
(LocationManager.GPS_PROVIDER);

if (location != null) {

latitude = location.getLatitude();

longitude = location.getLongitude();

}

}
```

```
}

}

}

} catch (Exception e) {

e.printStackTrace();

}

return location;

}
```

APPENDIX B

Sample code for creating homepage

```
<div class="row">
<br><br>
   <div class="col s12">
        <form class="col s12 l3 card push-l5 pad-login" action="<?php echo
        site_url('Bus/home');?>" method="post">
   <div class="input-field col s12">
       <select name="indata">
          <option value="" disabled selected>Choose your option</option>
          <option value="4">4corner</option>
          <option value="3">ecedept</option>
          <option value="2">demo2</option>
          <option value="1">demo1</option>
          <option value="5">KGISL</option>
          <option value="6">KCT</option>
          <option value="7">Kurumbapalayam</option>
       </select>
  <label>Materialize Select</label>
</div>
<div class="input-field right">
  <button class="btn waves-effect waves-light" type="submit"
name="action">Submit<i class="material-icons right">send</i>
  </button>
  </div>
</form>
</div></div>
```

APPENDIX C

Sample code for displaying time chart page

```
<div class="row container">
<br><br>
  <div class="col s12">
<table class="highlight">
  <thead>
    <tr>
        <th>Stop Name</th>
        <th>Bus Name</th>
        <th>Speed</th>
        <th>Time</th>
    </tr>
  </thead>
  <tbody>
      <?php foreach ($stat as $now) { ?>
    <tr>
      <td><?php echo $now->stopName ?></td>
      <td><?php echo $now->busId ?></td>
      <td><?php echo $now->speed ?></td>
      <td><?php echo $now->time ?></td>
    </tr>
    <?php } ?>
  </tbody>
</table>
</div>
</div>
```