COLLEGECODE:4212 **421221104050 PHASE4**

PublicTransportOptimizationusingIoTwithSensors

ProjectDescription:

Thisprojectaimstosolvetheproblemoftrackingandaccountabilityofvehiclesbyproviding a software platform. This project would serve as an important step to help in Vehicle tracking, component monitoring, vehicle analysis and fleet management. An efficientvehicle tracking system is implemented for monitoring of any equipped vehicle from any location at any time with the help of Global Positioning System (GPS) and Arduino Board which will enable users to locate their vehicles with ease and in a convenient manner.

GPSSensor:

Application:Usedforreal-timevehicletrackingandrouteoptimization.

Benefits:Providesaccuratelocationdata,helpingtomonitorvehiclemovement,calculateETA,and optimize routes based on traffic conditions.

Application: This project can be deployed in various fields like elderly and disabled care services,logisticdivision,emergencyservicesandrescueoperation,schoolbustracking,and accounting...

1. **Real-TimeTracking**
   Here, the GPS receiver receives the location data like latitude and longitude of a vehicle andsendsthembyusinganHTTPrequesttoweb server.Thebrowserisusedtoload the PHP web page which contains Google maps to show the location of the vehicle in real time. The web page containing the map directly marks the coordinates, as it arrives, without reloading the page. That means, in real time, we get to see the location of the vehicle.
2. **History**
   In history, we ask the user to select the date of journey, the names of vehicles which wereonajourneyonthat dayaredisplayedbyselectingthevehiclethejourneyofthat vehicle on that date is fetch. The google map with a marker is displayed on the screen. Along with the map, the Time-Speed Graph of journey and details of the journey is displayed. Inhistory,the usermustselectthedateofjourneyandcorrespondinglythe namesof vehicles active on that day are displayed. After the vehicle has been selected, corresponding journey details are displayed. Journey details include google maps representing journey, graph showing speed of vehicle at every instance of time and driver details consisting of driver details.
3. **ReportGeneration**
   Thispartincludesgenerationofspecificreportsfordrivers,trucks,andjourneys.Inthis

pdf report is generated, which are used by owners for analysis. For the driver's report, driver must be selected which then generates a report showing basic details and journeys done by him. For truck report, truck must be selected which then generates a report showing basic details and journeys done by him. For journey report, date and vehicleneedtobeselectedwhichthengenerates areportshowingbasicjourneydetails, google maps representing complete journey and graph showing speed of vehicle at every instance of time.

4. **Geofencing**

   Geofencing is one of the important features of our project. As the problem statement stated that driver may take a longer route to increase fuel consumption and ask for moremoney forhis longjourney orthedriverunintentionallytookthewrongpath,such scenario affects the throughput of the journey. It also affects the performance of vehicles and drivers. Thus, geofencing is applied.

   Geofencing is the virtual boundary to the optimized path from source to destination whichisdefinedatthetime ofjourney creation.Itisaradiuswhichthe userhastoinput inmeters. Whenevervehicle deviates from its optimizedpath andthe deviationis more than the defined radius then it means it’s a geofencing break, which results in a notificationpopupgeneratedonthescreen.This helpstheusertoknowthatgeofencing is broken by this vehicle.

5. **SMS Module**

   SMS module is an important feature for the user. As the project is for vendors and they are very busy they can’t stay in front of the computer and check the current location of the vehicle. In this scenario, the SMS module is useful to know the current location of vehicle. To know the current location of the vehicle user, you must send an SMS to several of the sims attached to the hardware. In response, the SMS a link is generated whenuserclicksonthatlinkonthebrowseritshowsagooglemapwiththemarker.The marker is the current location of the vehicle at that time. Thus, the SMS module helps the user to check the location of vehicle anytime he wants.

6. **AgileHardware**

   Hardware isreliable,lowcostand compact.Aswestatedthatthisprojecthasvarious applications invariousareassoforthoseapplications'hardwarewillbe fixedonly,we must change the web application and the method to use the functionality. The functionality and construct of this hardware is fixed and working for this.

Languages: C,PHP,JavaScript,Ajax,Bootstrap,HTML,CSS
Hardware: ArduinoUNO,SIM808,GPS antenna

DEVELOPINGTHEREAL-TIMETRANSITINFORMATIONPLATFORM BYUSINGWEBTECHNOLOGY

```
<htmllang="en">
<head>
  <metacharset="UTF-8">
  <metaname="viewport"content="width=device-width,initial-scale=1.0">
  <title>Real-TimeTransitInformation</title>
  <linkrel="stylesheet"href="styles.css">
</head>
<body>
  <header>
    <h1>Real-TimeTransitInformation</h1>
  </header>
  <main>
    <div id="map"></div>
    <div id="schedule">
      <h2>UpcomingBuses</h2>
      <ulid="bus-schedule">
        <!--Real-timebusscheduledatawillbedisplayedhere-->
      </ul>
    </div>
  </main>
  <scriptsrc="script.js"></script>
</body>
</html>
```

CSS

```
body{
  font-family:Arial,sans-serif; margin:
  0;
  padding:0;
}
```

Header

```
{
  background-color:#333;
  color: #fff;
  text-align:center;
  padding: 20px;
}
Main
{
  display: flex;
  justify-content:space-between;
  padding: 20px;
}
{
  flex:
  1;height:400p
  x;
}
{
  flex: 1;
  padding:10px;
}

#bus-schedule{
  list-style-type:none;
  padding: 0;
}
{
  margin:5px0;
  border:1pxsolid#ccc;
  padding: 10px;
}
```

Javascript

```
document.addEventListener("DOMContentLoaded",()=>{ const
  busSchedule = [
    {route:"Bus101",destination:"Downtown",nextArrival:"10:15AM" },
    {route:"Bus202",destination:"Suburbia",nextArrival:"10:30AM"},
    {route:"Bus303",destination:"Airport",nextArrival:"10:45AM" },
  ];
  constbusScheduleList=document.getElementById("bus-schedule"); busSchedule.forEach((bus)
  => {
    constlistItem= document.createElement("li");
    listItem.innerHTML=`<strong>${bus.route}</strong>to${bus.destination}<br>Nextarrival:
${bus.nextArrival}`;
    busScheduleList.appendChild(listItem);
  });
});
```

DESIGNTHEPLATFORMTORECEIVEANDDISPLAYREAL-TIMEDATAFROMIOT SENSORS: