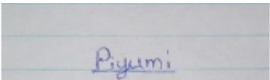


Name: Elgiriya Prasandika

Student Reference Number: 10749850

Module Code: PUSL2020	Module Name:
Coursework Title: PUSL2020 Group Assignment (Team 1)	
Deadline Date: :19/05/2022	Member of staff responsible for coursework: Dr. Rasika Ranaweera
Programme: BSc (Hons) Computer Science	
Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook .	
Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.	
1.Elgiriya Prasandika	10749850
2.Swrnathilaka Amarajeewa	10749949
3.Asalage Navodya	10749851
4.Nishshankage Nishshanka	10749846
5.Egallalage Nandasiri	10749855
6.Mahadurage Abeyrathna	10749852
<i>We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.</i>	
Signed on behalf of the group:	
Individual assignment: <i>I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.</i>	
Signed:	
Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.	
I *have used/not used translation software.	
If used, please state name of software.....	
Overall mark ____%	Assessors Initials _____ Date_____

PUSL 2020 **SOFTWARE**
DEVELOPMENT
TOOLS AND PRACTISES

PROPOSED BY: GROUP 01

Final Report

Lecturer- Dr. Rasika Ranaweera

Submission Date: 19/05/2022

Acknowledgement

First, we would like to express our heartfelt gratitude to our module leader as well as the Dean of Computing Faculty, Dr. Rasika Ranaweera sir and Mrs. Pavithra Subhashini madam for guiding us for the successful of our project. In this pandemic situation as lecturers, they step forward and pushed forward us for educational activities. We are very thanking full to our module lectures for theirs' patience, motivation, enthusiasm, and vast knowledge. The advice given by the lectures in charge for the module PUSL 2020 Software development tools and practices are more invaluable and more helpful to create this project and the report for success.

During this process we also very thankful to Mr. Pramudhya Thilakarathna sir for giving us guidance to this module by conducting practical sections throughout the semester. It was very helpful to us to create this website.

We are also very thankful to NSBM Green University and for Plymouth University for providing us these kinds of educational opportunities.

At last, I am very thankful to our group members for the work that they have done during this project. It is very incredible and very memorable for all of us. Not only them but also, we are very thankful to our parents, academic staff and for the nonacademic staff for all their advice, guidance and for the help.

Group No:01
BSc (Hons) Software Engineering
Nsbm Green University
Plymouth university.

Abstract

This project is regarding a website which has been created to store, identify, and take quick actions for relevant situation during an accident. This web site can be used for all the police department, insurance companies all the vehicle owned drivers and for the RDA department.

For this application, all the drivers those who own a vehicle can register to the web application by supplying necessary details. RDA officers can also log in to the web application by using the details same as insurance officers. Both police and the insurance offices logging to the web application by same dashboard. All the drivers those who have accounts can add, delete accident details. Both police and insurance officers can see and respond to them.

Police officers, insurance offices and the RDA officers can accept or reject the claim that is send by the drivers. Police officers can see the relevant accident type also by using the graphs.

This application is very user friendly, and we have used PHP, HTML, CSS, Bootstrap, JavaScript for creating this project. We have use PHP for connecting the databases. All ages people can use this application. It is very easy to understand, and user guidance is given.

Table of Contents

01.Chapter 01	7
1.1 Introduction.....	7
1.2 Scope of the project	7
02.Chapter 02	8
2.1 Website Requirements	8
2.1.1. Functional requirements	8
2.1.2 Non-Functional requirements.....	8
2.2. Features of the website	9
03.Chapter 03	9
3.1 Site Map	9
04.Chapter 4	10
4.1 Software and Tools.....	10
4.2 Programming Languages	10
05.Chapter -Testing	11
5.1 Unit tests and integration tests	11
5.2 Functional testing	11
5.2.1. Functional Test plan.....	11
5.4 Test and validation metrics.....	14
5.4.1. Test plan	14
5.4.1. Code of validation	18
06. Future Implementation	25
07.Summary.....	26
08.Conclusion.....	26
09. Appendices	26
9.1.Pages of the web site	26
09.Team contribution	37
9.1 Details of members.....	37
9.2 Contribution Statement of group members.....	38

Table of figures

Figure 1:Site map of the web application	9
Figure 2:Functional Testing 01.....	12
Figure 3:Functional Testing 02.....	12
Figure 4:Functional Testing 03.....	12
Figure 5:Functional Testing 04.....	12
Figure 6:Functional Testing 05 fail.....	13
Figure 7:Functional testing 05 pass	13
Figure 8:Functional Testing 06.....	13
Figure 9:Driver login validation.....	15
Figure 10:driver sign up Validation.....	15
Figure 11:Add accident Validation	16
Figure 12:Police, RDA, Insurance Login validation	16
Figure 13:Polic, RDA, Insurance Login validation unsuccessful	17
Figure 14:Add accident validation	18
Figure 15:Chart validation 01	19
Figure 16;Chart Validation 02	20
Figure 17:Line chart	21
Figure 18:Driver account create validation.....	22
Figure 19:Driver sign In Validation	23
Figure 20;Login Check-up validation.....	23
Figure 21;Login validation	24
Figure 22:Police form Validation	24
Figure 23:Insurance table Validation.....	24
Figure 24;Log out Validation	25
Figure 25>Delete account Validation	25
Figure 26:Home page 01	26
Figure 27:Home page 02	27
Figure 28;Home page 03	28
Figure 29:Main Accitex	28
Figure 30:Register page	28
Figure 31:Gallery	29
Figure 32:Sponsor	29
Figure 33:Login	30
Figure 34:Login failed	31
Figure 35:Insurance login detail	31
Figure 36:login successful	31
Figure 37:RDA login.....	32
Figure 38:Charts 01	33
Figure 39:Charts 02	33
Figure 40:Accitex data table 01.....	33
Figure 41:Accitex data table 02.....	34
Figure 42:Driver Sign In page	34
Figure 43:Accident report page.....	35
Figure 44:Create account page Driver	35
Figure 45:Recorded accident page	36
Figure 46:Add accident page	36

01.Chapter 01

1.1 Introduction

This project is regarding a web application developed by our team members. The RDA of Sri Lanka along with the Police and Insurance Companies plan to launch a website to track traffic accidents. This application is very user friendly, and guidance is provided.

The website will provide a platform for the drivers to report accidents and traffic incidents. The webmaster will create the accounts for Insurance Companies, Sri Lankan Police and to the RDA. Drivers can login or sign up to the site by themselves. Using the vehicle's registration number and insurance information, anyone who owns or operates a vehicle can register with the system. When an accident occurs, a driver can report it via the website, along with supporting images. The website will provide a platform for the drivers to report accidents and traffic incidents. The webmaster will create the accounts for Insurance Companies, Sri Lankan Police and to the RDA. Drivers can login or sign up to the site by themselves. RDA officers, police, and applicable insurance firms can inspect all reported incidences, validated them by examining the submitted images, and take appropriate action such as approving or denying the claim. From this website the police can view approved accidents as various graphs, such as percentages of bicycle, automobile, bus, etc. accidents or accident reasons.

To develop this application, we have used main software languages such as PHP, JavaScript, CSS, Bootstrap, and Html. We used a MySQL database server to store databases.

This web application is very easy to use and anybody can use this application.

1.2 Scope of the project

Our project is developing a website for our client The Road Development Authority (RDA) of Sri Lanka along with the Sri Lankan Police and Insurance Companies. At first team will analyze the client Requirements. This website will let the drivers report accidents so Police can track them.

This website is accepted to make the traffic accident management process efficient. It is accepted to have web master who can create accounts for RDA and police staff. Drivers should be able to sign up for free and input data and update. Police should be allowed to see the reported incidents and approve them. Also, they should be delivered with graphs to analyze accidents when needed. We are accepted To deliver PHP based web application with MySQL database.

02.Chapter 02

2.1 Website Requirements

Website requirements is a list that contain necessary functions, capabilities, or characteristics related to the website that we have created. There are many types of requirements that may be defined during the process that come together to focus and prioritize the project plan.

- Business requirements
- User requirements
- Security requirements
- Functional requirements
- Non-functional requirements

In here we are discussing about the functional and non-functional requirements.

2.1.1. Functional requirements

Functional features are the product features the developers must implement to enable to accomplish their task for the users. This describes the system behavior under specific conditions.

- There should be a "Web Master" who can create accounts for Police, RDA, and insurance employees.
- Registration is free for drivers and can register by themselves
- Registered drivers can add information (including photographs), delete, and update their own occurrences.
- Police, Insurance and RDA can view a list of reported instances.
- Insurance and Police can click to expose additional facts, then approve or refuse them.
- The police can also view the graphs and analyses of accident types and reasons.

2.1.2 Non-Functional requirements

These features are not directly dependent on the users. It means the system can exist without non-functional requirements.

- Security
- Appearance
- Privacy
- Dark or light themes
- Performance of the web application
- Quick response from police, RDA, and insurance .
- Supportability

2.2. Features of the website

- Well designs and functional
- Easy to use
- User friendly
- User information Security
- View Accident details
- Related accident photos
- User logging (drivers)
- Personalized items
- Drivers reviews
- Security options
- Dashboard
- Content management
- Tracking accidents
- Automatic backups

03.Chapter 03

3.1 Site Map

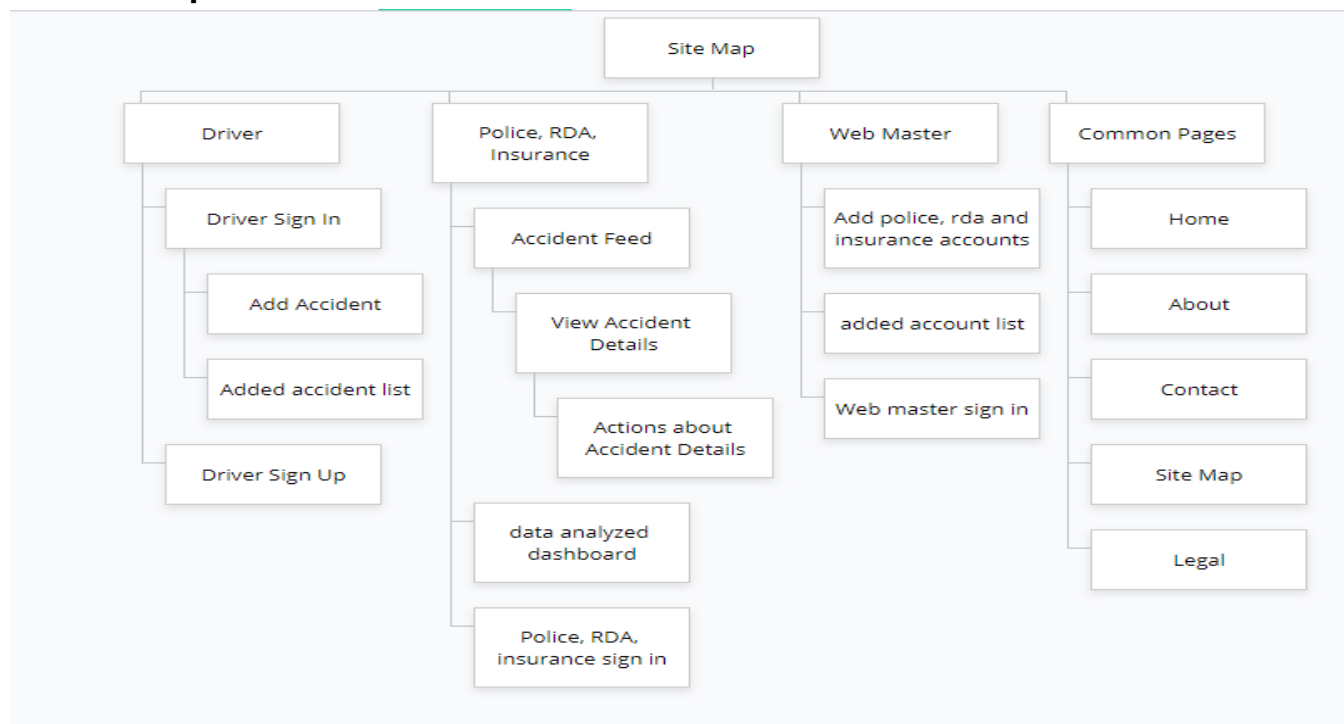


Figure 1:Site map of the web application

04.Chapter 4

4.1 Software and Tools

There are many software tools that are now available in information technology. Those are very helpful for the Back end and front-end developers in computing section. By those we also have used many technologies. Some of them are given below:

- Sublime Text
- GitHub
- Bootstrap
- Brackets
- Visual Studio Code
- W3 School
- Chartjs

4.2 Programming Languages

There are many language tools in nowadays. Due to developing of world, many new languages and tools are invented. There are so many languages such as Java, Python, C#, C, PHP ...etc. To develop this web application we have used PHP,HTML, CSS, JavaScript, Bootstrap. We have use MySQL database to store data in the web application.

Here are some popular web application tools that might assist us for creating this:

- Automation of text
- Wearable web application Tools
- Online Video for guidance
- Responsive Design
- Application Programming Interface
- Analytics
- A More Secure Cloud security system

05.Chapter -Testing

5.1 Unit tests and integration tests

Unit testing is a sort of software testing that examines individual software units or components. The goal is to ensure that each unit of software code works as intended. Developers perform unit testing throughout the development (coding) phase of an application. Unit tests are used to isolate a part of code and ensure that it is correct. A singular function, method, procedure, module, or object might be considered a unit.

5.2 Functional testing

Testing for functional requirements is a type of software testing. Functional tests check that the output matches the functional requirements by doing tests on each individual function of the software application.

Most of the functional testing is done in a "black box" method, and it is not concerned with the application's source code. The user interface, APIs, databases, security, and client-server communication are all examined as part of this testing. Manual testing or automated testing are both viable options.

5.2.1. Functional Test plan

Test Case ID	Test case name	Expected result
W001	Create Accounts to Police, Insurance and RDA	Add insurance, police, RDA data into database and provide them username and password.
W002	Registration is free for drivers and can register by themselves.	Driver can register free and login to driver dashboard.
W003	Registered drivers can add information (including photographs), delete, and update their own occurrences	Driver can add accident details, delete details add update details.
W004	Police, Insurance and RDA can view a list of reported instances.	Police/RDA/ Insurance can view reported accidents.
W005	Insurance and Police can click to view additional facts, then approve or refuse them.	Can view/delete/accept relevant accident details.
W006	The police can view the graphs and analyses of accident types and reasons.	Police can view causes of accident type chart and accident vehicle type chart.

Test 01

Web master								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W001	Creates accounts to the Police	Webmaster should sign upto the admin panel	login to the admin panel	Username="M_police" Password="m123"	Add insurance, police, RDA data into database and provide them username and password	As expected Insurance and police can login by using provided username and password	Pass	
	Create accounts to the Insurance		Choose a relevant form to fill police/insurance/RDA data	Username="piyu" Password="123"				
	Create accounts to the RDA		Give suitable username and password					

Figure 2:Functional Testing 01

Test 02

Driver registration								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W002	Registration is free for drivers and can register by themselves	should have vehicle registration numbers and insurance	login to ACCITEX website.	name="", nic="", tel="", licnum="", unname="", pw=""	Driver can register free and login to driver dashboard	As expected driver can register free and login to driver dashboard	Pass	
			Then go to driver login page					
			Select sign up					
			Add required details and register					

Figure 3:Functional Testing 02

Test 03

Add Accident and delete data								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W003	Registered drivers can add information (including photographs), delete, and update their own occurrences	Should register as driver	Login to the driver dashboard	Accident Vehicle="car", Vehicle No="GT123", Reason="Drunk Driving", location="Homagama", Add images="pic_1"	Driver can add accident details, delete details add update details	As an expected driver can add accident details and delete data	Pass	
			Click Add Accident					
			Add accident data and relavent photos					
			Delete or update data					

Figure 4:Functional Testing 03

Test 04

View a list of reported accidents								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W004	Police, Insurance and RDA can view a list of reported instances	should have access to the police and insurance dashboards	Login to the police/insurance dashboard	username="ceylinco" password=""cey123	Police/RDA/ Insurance can view reported accidents	As expected Police and Insurance can view reported accidents	Pass	

Figure 5:Functional Testing 04

Test 05

Insurance and Police can click to view additional facts, then approve or refuse them								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W005	Insurance and Police can click to view additional facts, then approve or refuse them	should have access to the police and insurance dashboards	Login to the police/insurance dashboard	Data from add accident table	Can view/delete/accept relevant accident details	As expected insurance can view/ delete or accept accidents but can't police cant take actions	Fail	
			Click view/accept/delete					

Figure 6:Functional Testing 05 fail

Insurance and Police can click to view additional facts, then approve or refuse them -Test Round 02								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W005	Insurance and Police can click to view additional facts, then approve or refuse them	should have access to the police and insurance dashboards	Login to the police/insurance dashboard	Data from add accident table	Can view/delete/accept relevant accident details	As expected insurance and police can view/ delete or accept accidents	Pass	
			Click view/accept/delete					

Figure 7:Functional testing 05 pass

Test 06

The police can view the graphs and analyses of accident types and reasons								
Test Case ID	Test Description	Prerequisite	Test Steps	Input Data	Expected Data	Actual Result	Status	Comment
W006	The police can view the graphs and analyses of accident types and reasons	should have access to the police dashboards	Login police dashboard		Police can view causes of accident type chart and accident vehicle type chart	As expected police can see relevant pie chart and bar chart	Pass	
			view charts					

Figure 8:Functional Testing 06

5.4 Test and validation metrics

5.4.1. Test plan

Test Case ID	Test Case name	Expected result
W_Login_001	Verify where admin can login successfully	Login to the admin dashboard
W_Login_002	Verify insurance and police can login successfully	Login to the insurance and police dashboard
W_Login_003	Verify drivers can login their accounts successfully	Login to driver dashboard
W_Login_004	Verify where correct user is getting login or not	Should display an error message and return to the login page
W_webmas_005	Add Details to the insurance and police table	Should display insurance and police details in the relevant table
W_Insurance_006	Display accident details	Should display accident details in insurance dashboard
W_Insurance_007	Insurance can update of reject accidents. Then approved accidents goes to approved accident details.	Should display accepted accidents in the approved accident table.
W_Insurance_008	Rejected accidents remove from the insurance dashboard	Should remove from the data which reject by the insurance company
W_Police_009	Police can see accident cause through a pie chart and accident vehicles through the bar chart	Update the percentage of the cause of the accident and accident vehicles from the bar chart
W_Police_010	Police can see number of accepted and rejected accidents	Should Update number of accidents and rejected accidents
W_driver_011	Submit accident details	Should display accident details in the driver dashboard
W_driver_012	By clicking the logout button user should navigate to login page	Should redirect to login page when click the logout button
W_navigation_013	Navigation bar	Use them to move to different pages



Test cases (2).xlsx

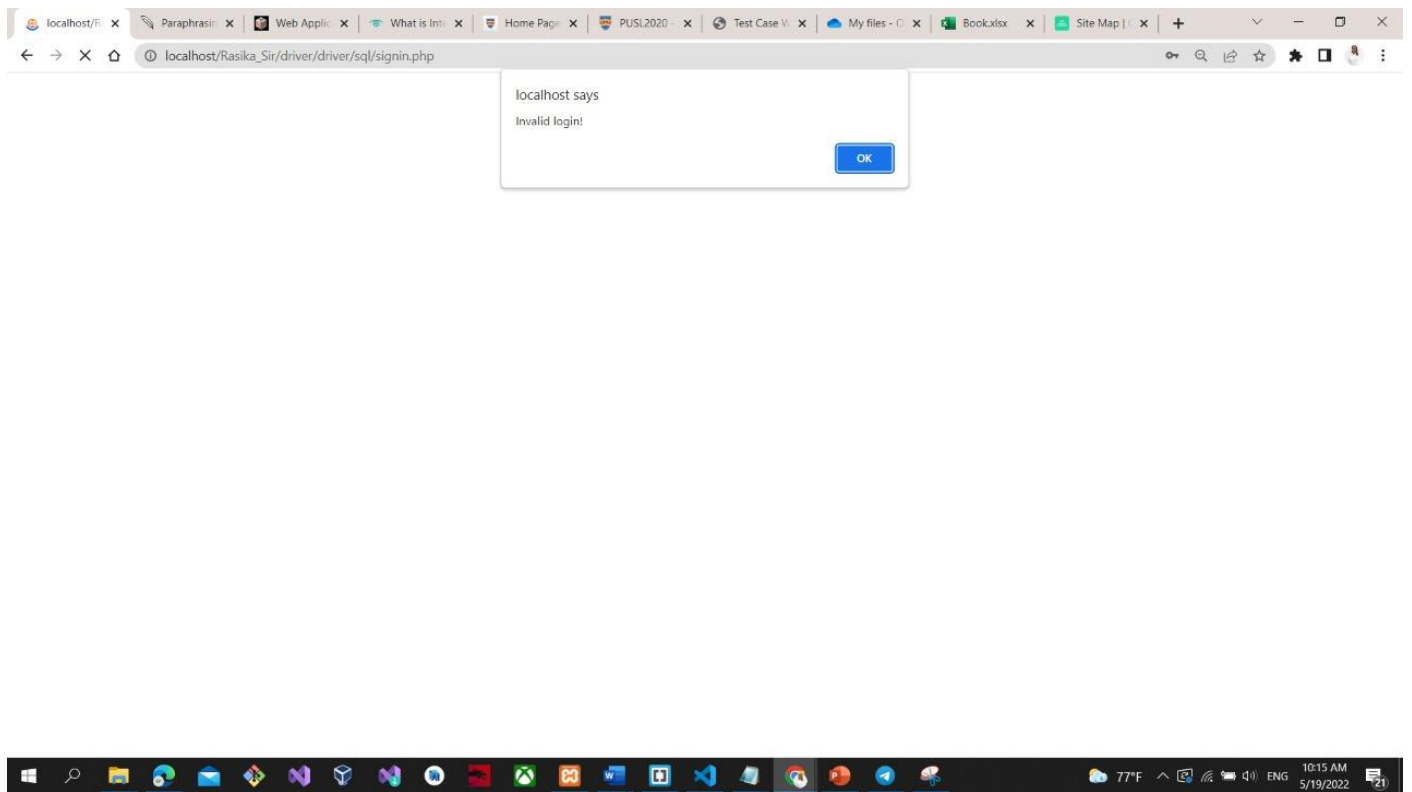


Figure 9:Driver login validation

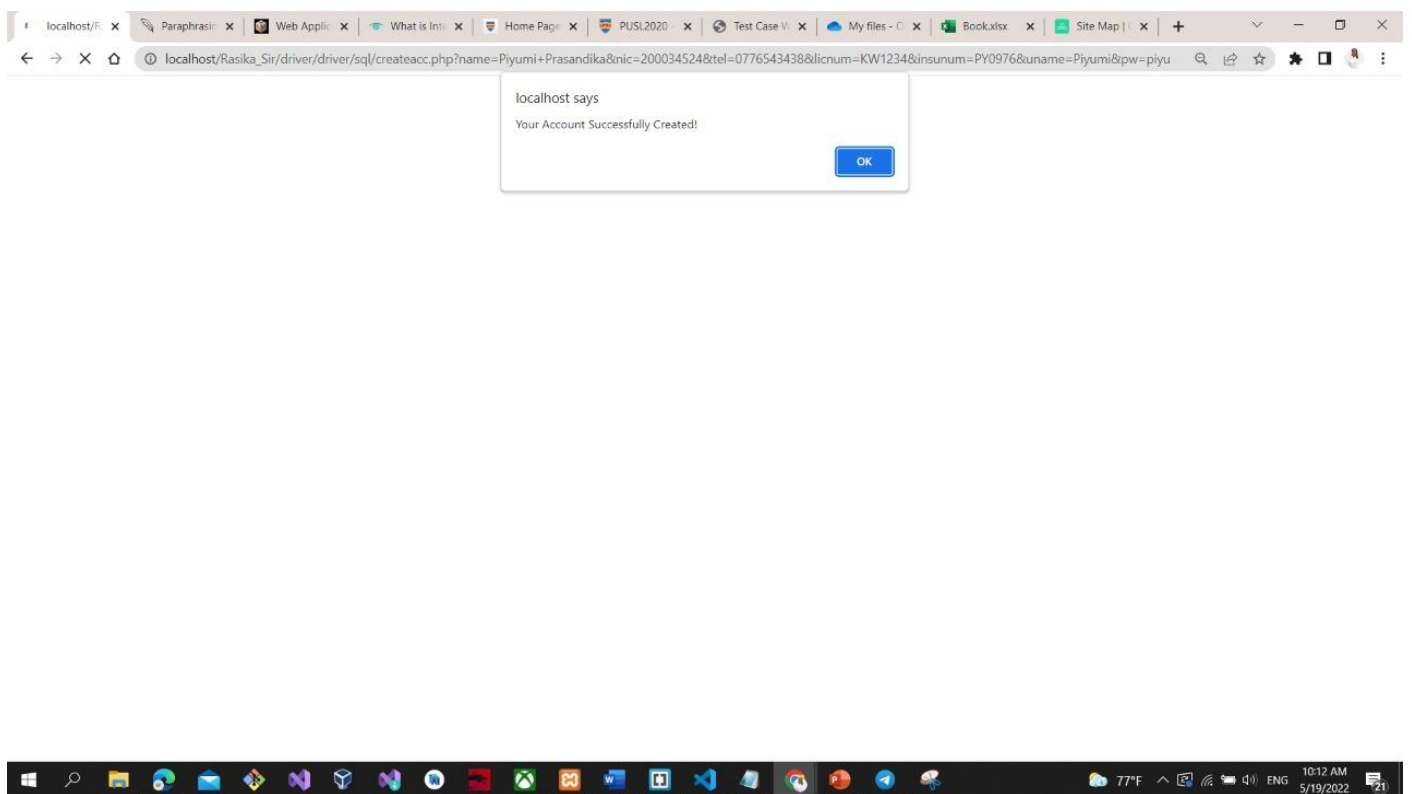


Figure 10:driver sign up Validation

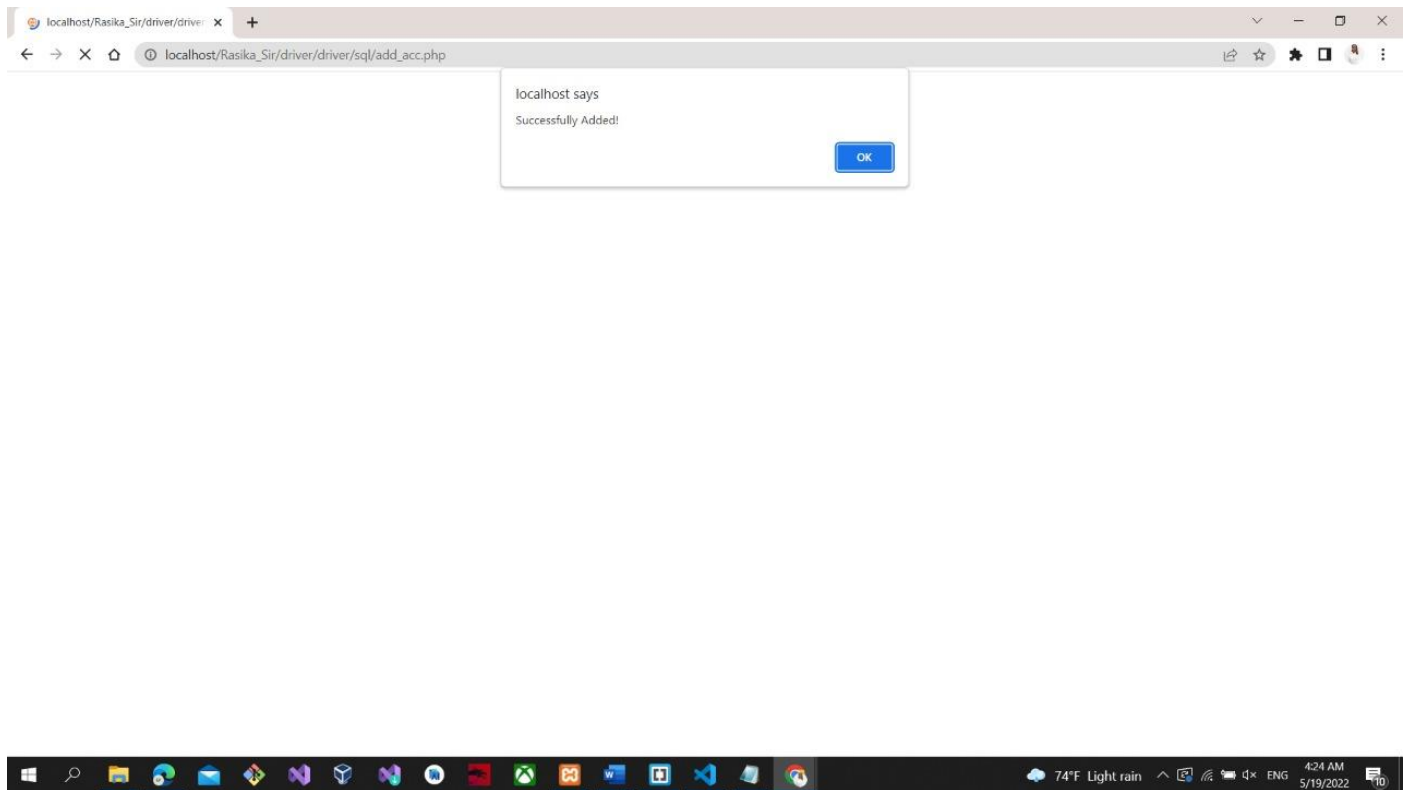


Figure 11: Add accident Validation

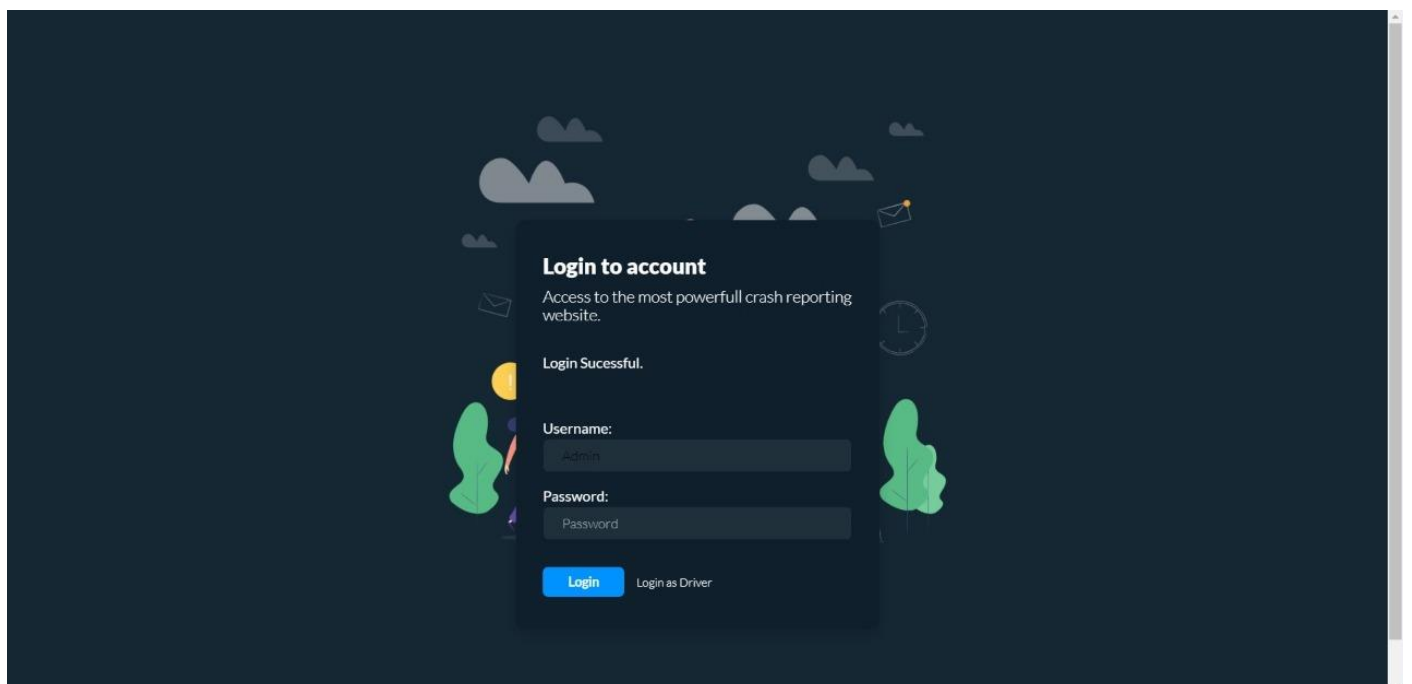


Figure 12: Police, RDA, Insurance Login validation

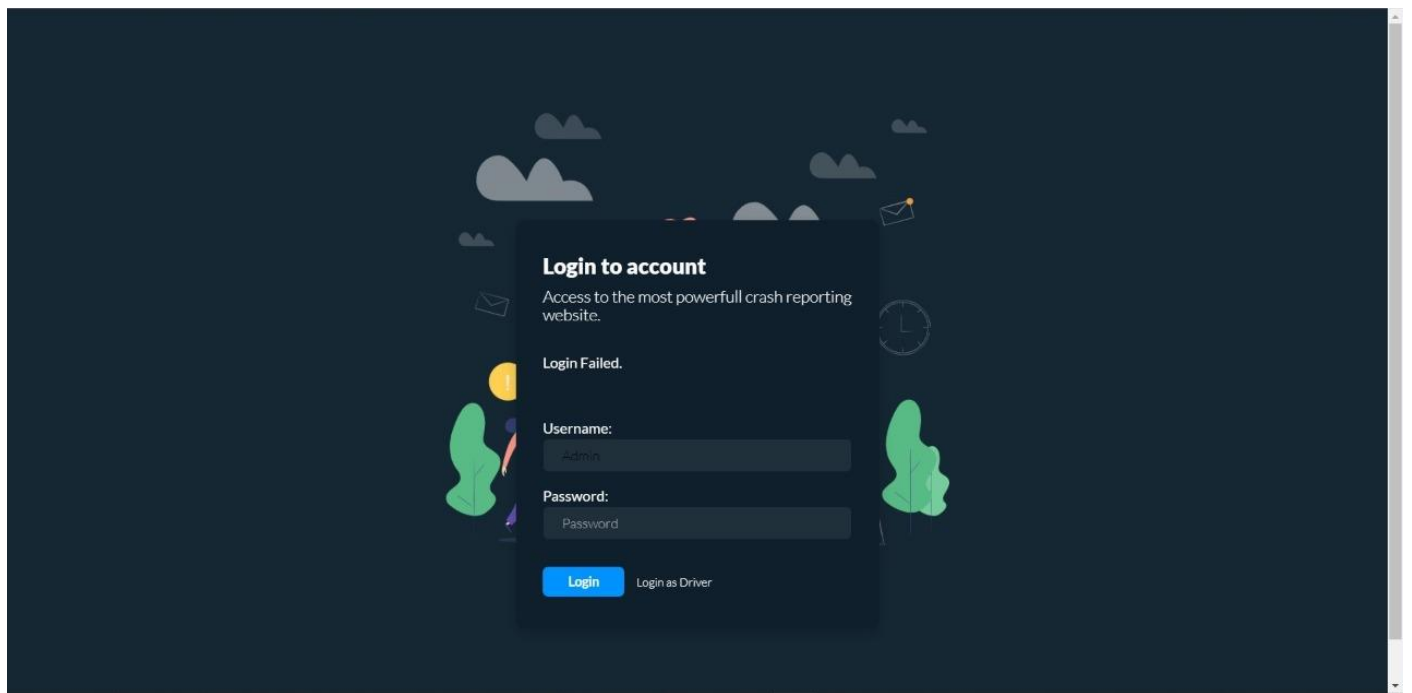


Figure 13: Polic, RDA, Insurance Login validation unsuccessful

5.4.1. Code of validation

```
<?php
    $vehi=$_REQUEST["vehi"];
    $source=$_REQUEST["source"];
    $des=$_REQUEST["des"];
    $acctype = $_REQUEST["acctype"];
    $date=date("Y-m-d H:i:s");
    $did=$_SESSION["driver"];

    //image 1
    $file_tmp1 = $_FILES['pic1']['tmp_name'];
    $file_name1 = "v"."_" .rand(1,100)."_" .$_FILES['pic1']['name'];
    echo $target_file1 = "img/" . $file_name1;
    if($file_tmp1!="")
    { move_uploaded_file($file_tmp1,$target_file1); }
    else { $file_name1=""; }

    //image 2
    $file_tmp2 = $_FILES['pic2']['tmp_name'];
    $file_name2 = "v"."_" .rand(1,100)."_" .$_FILES['pic2']['name'];
    $target_file2 = "img/" . $file_name2;
    if($file_tmp2!="")
    { move_uploaded_file($file_tmp2,$target_file2); }
    else { $file_name2=""; }

    //image 3
    $file_tmp3 = $_FILES['pic3']['tmp_name'];
    $file_name3 = "v"."_" .rand(1,100)."_" .$_FILES['pic3']['name'];
    $target_file3 = "img/" . $file_name3;
    if($file_tmp3!="")
    { move_uploaded_file($file_tmp3,$target_file3); }
    else { $file_name3=""; }

    $sql="INSERT INTO acc(vehicle,source,des,did,date,pic1,pic2,pic3,type)
    VALUES('$vehi','$source','$des','$did','$date','$file_name1','$file_name2','$file_name3','$acctype)";

    //Execute command 2
    if ($conn->query($sql) === TRUE) {
        echo "<script>alert('Successfully Added!'); window.location.href='../acc.php' ;</script>";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
?>
```

Figure 14:Add accident validation

```

<?php
$sql="SELECT type,count(type) as total FROM acc group by type";
$result = $con->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {

        if($row["type"]=="Bad Weather"){
            $Bad_Weather=$row["total"];
        }
        elseif($row["type"]=="Drunk Driving"){
            $Drunk_Driving=$row["total"];
        }
        elseif($row["type"]=="Speeding"){
            $Speeding=$row["total"];
        }
        elseif($row["type"]=="Distractions"){
            $Distractions=$row["total"];
        }

    }

}

$sql="SELECT vehicle,count(vehicle) as total FROM acc group by vehicle";
$result = $con->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {

        if($row["vehicle"]=="jeep"){
            $jeep=$row["total"];
        }
        elseif($row["vehicle"]=="Van"){
            $van=$row["total"];
        }
        elseif($row["vehicle"]=="Bike"){
            $bike=$row["total"];
        }
        elseif($row["vehicle"]=="Car"){
            $car=$row["total"];
        }

    }

}

```

Figure 15:Chart validation 01

```

<script>

const dataP = {
  labels: [
    'Bad Weather',
    'Drunk Driving',
    'Speeding',
    'Distractions'
  ],
  datasets: [{
    label: 'My First Dataset',
    data: [
      <?php echo $Bad_Weather; ?>,
      <?php echo $Drunk_Driving; ?>,
      <?php echo $Speeding; ?>,
      <?php echo $Distractions; ?>
    ],
    backgroundColor: [
      'rgb(255, 99, 132)',
      'rgb(54, 162, 235)',
      'rgb(255, 205, 86)',
      'rgb(75, 192, 192)'
    ],
    hoverOffset: 4
  }]
};

const configP = {
  type: 'doughnut',
  data: dataP,
};

// Line chart
const dataL = {
  labels: ["Car", "Jeep", "Bus", "Van", "Bike"],
  datasets: [{
    label: 'Accident details charts',
    data: [
      <?php echo $car; ?>,
      <?php echo $jeep; ?>,
      <?php echo $bus; ?>,
      <?php echo $van; ?>,
      <?php echo $bike; ?>
    ]
  }]
};

```

Figure 16;Chart Validation 02

```

// Line chart
const datal = {
  labels: ["Car","Jeep","Bus","Van","Bike"],
  datasets: [{
    label: 'Accident details charts',
    data: [
      <?php echo $car; ?>,
      <?php echo $jeep; ?>,
      <?php echo $bus; ?>,
      <?php echo $van; ?>,
      <?php echo $bike; ?>
    ],
    backgroundColor: [
      'rgba(255, 99, 132, 0.2)',
      'rgba(255, 159, 64, 0.2)',
      'rgba(255, 205, 86, 0.2)',
      'rgba(75, 192, 192, 0.2)',
      'rgba(54, 162, 235, 0.2)',
    ],
    borderColor: [
      'rgb(255, 99, 132)',
      'rgb(255, 159, 64)',
      'rgb(255, 205, 86)',
      'rgb(75, 192, 192)',
      'rgb(54, 162, 235)',
    ],
    borderWidth: 1
  }]
};

const configl = {
  type: 'bar',
  data: datal,
  options: {
    scales: {
      y: {
        beginAtZero: true
      }
    }
  },
};
</script>

```

Figure 17:Line chart

```

<?php include 'conn.php'; ?>
<?php
$name=$_REQUEST["name"];
$licence=$_REQUEST["licnum"];
$nic=$_REQUEST["nic"];
$tel=$_REQUEST["tel"];
$username=$_REQUEST["uname"];
$password=$_REQUEST["pw"];
$insu=$_REQUEST["insunum"];

$sql="INSERT INTO driver(dname,dnic,dtel,demail,uname,pw,insu)
VALUES('$name','$nic','$tel','$licence','$username','$pw','$insu')";

//Execute command 2
if ($conn->query($sql) === TRUE) {
    echo "<script>alert('Your Account Successfully Created!'); window.location.href='../index.php' ;</script>";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
?>

```

Figure 18:Driver account create validation

```

<?php
include 'conn.php';
$username=$_REQUEST["uname"];
$password=$_REQUEST["pw"];

$login_check=0;

$sql=" SELECT * FROM driver WHERE uname='$username' AND pw='$pw' ";
$result=$conn->query($sql);
while($row=$result->fetch_assoc())
{
    $login_check=1;
    $did=$row["did"];
}

if($login_check==1)
{
    $_SESSION["driver"]=$did;
    header('Location: ../acc.php'); exit;
}
else
{
    echo "<script>alert('Invalid login!'); window.location.href='../index.php' ;</script>";
}
?>

```



```

<?php
//Authorization -Access control
//Check whether the user is logged in or not
if(!isset($_SESSION['user']))
{
    //user is not logged in
    //Redirect to login page with message
    $_SESSION['no-login-message']="

>Please login to access Admin Panel.</div>";

    //Redirect to login page
    header('location:'.SITEURL.'/admindash/login.php');
}
?>


```

Figure 20;Login Check-up validation

```

$message = "";

//check whether the submit button is clicked or not
if(isset($_POST['submit']))
{
    $username = $_POST['username'];
    $password = $_POST['password'];

    $hashed_password = md5($password);

    echo $username;
    echo $password;
    //2.SQL to check whether the user with username and password exists or not
    $sql="SELECT * FROM users WHERE username='$username' AND password='$password'";

    //3.Execute the Query
    $res = mysqli_query($conn, $sql);

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // output data of each row
        while($row = $result->fetch_assoc()) {
            $utype=$row["user_type"];
            $uname=$row["username"];
            $pwd=$row["password"];
        }

        $_SESSION['user_type']=$utype;

        //user available and login success
        $_SESSION['login']="

>Login Successful.</div>";
        $_SESSION['user']=$uname; //To check whether the user is logged in or not and logout will unset it

        if($password==$pwd){
            if($utype=="admin"){echo"<script> window.location = 'admindash.php'; </script>";}
            elseif($utype=="insurance"){echo"<script> window.location = 'insurancedash.php'; </script>";}
            elseif($utype=="police"){echo"<script> window.location = 'policedash.php'; </script>";}
        }

    }
    else{
        $_SESSION['login']="

>Login Failed.</div>";
    }
}


```

```

        <?php
        if (isset($_POST['policesub'])){
        $poliareaname=$_POST['areaname'];
        $polid=$_POST['polid'];
        $poliemail=$_POST['poliemail'];
        $Contact_No=$_POST['contact'];
        $poliuname=$_POST['poliusername'];
        $polipwd=$_POST['polipassword'];
        $sql = "INSERT INTO tbl_police VALUES ('$poliareaname','$polid','$poliemail','$Contact_No','$poliuname','$polipwd')";

        if (mysqli_query($con, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($con);
        }

        $sql = "INSERT INTO users VALUES ('$poliuname', '$polipwd', 'police')";

        if (mysqli_query($con, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($con);
        }

    }

```

Figure 22: Police form Validation

```

        </div>
        <!-- end row -->
        <?php
        if (isset($_POST['insurance'])){
        $comname=$_POST['comname'];
        $tegid=$_POST['tegid'];
        $email=$_POST['email'];
        $regnamedate=$_POST['regnamedate'];
        $tel = $_POST['tel'];
        $i_username=$_POST['username'];
        $i_password=$_POST['password'];

        $sql = "INSERT INTO insurance VALUES ('$comname', '$tegid', '$email','$regnamedate','$tel','$i_username','$i_password')";

        if (mysqli_query($con, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($con);
        }

        $sql = "INSERT INTO users VALUES ('$i_username', '$i_password', 'insurance')";

        if (mysqli_query($con, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . "<br>" . mysqli_error($con);
        }

    }

```

Figure 23: Insurance table Validation

```

<?php
//Include constants.php for SITEURL
include('../config/constants.php');
//1.Destroy the session
session_destroy();//unset $ _SESSION['user']

//2.Direct to login page
header('location:'.SITEURL.'admindash/login.php');
?>

```

Figure 24;Log out Validation

```

<?php
include 'conn.php';
$accid=$_REQUEST["accid"];
// sql to delete a record
$sql = "DELETE FROM acc WHERE accid=$accid";

if ($conn->query($sql) === TRUE) {
    echo "<script>alert('Successfully Deleted!'); window.location.href='../acc.php' ;</script>";
} else {
    echo "Error deleting record: " . $conn->error;
}

?>

```

Figure 25:Delete account Validation

06. Future Implementation

In future we hope to develop this system in many sites by adding ongoing technologies, tools, and languages. In here we have used PHP, MySQL, and HTML our main languages. In future we are trying to change this and trying to make an advanced database system. Below some points are describe the future implementations of our application.

- **Add An Online Payment Method**

By this feature the insurance and the drives can pay necessary payments for near Police station, Post office and drivers claim.

- **Develop The Add Accident Details**

From this feature, the officers can again a full clear, and explained details of an accident immediately.

- **Develop Profile Feature of The User**

From this, the officers can view driver personal documents such as Vehicle Register Documents, Insurance details and Licen details.

- **Add Contac Numbers of Necessary Parties**

This feature is added for the help of drivers, police, insurance, and RDA officers. By this, they can easily get the contact number of near departments.

- **Detect Live Location of Drivers**

07.Summary

In this project we have discussing about a website that we have created for track traffic accidents. Here police officer, insurance offices and RDA officers are the users. By this driver can update delete and edit the details all the officers can view , accept, and reject the claim. This is very user-friendly application and PHP, MySQL, CSS, HTML, JavaScript and Bootsrap is used for constructing this. Any vehicle owned drivers can register to the application by Inputting Vehicle registered Number and Insurance details. Police, Insurance and RDA officers can also log in to the system and username and password is given by the web master

08.Conclution

As a group we work hard for the success of our project. As a final result, now have completed our task successfully. We are hope t develop this application furthermore with ongoing technical and tools. After that it may be very easy to use and control

09. Appendices

9.1.Pages of the web site

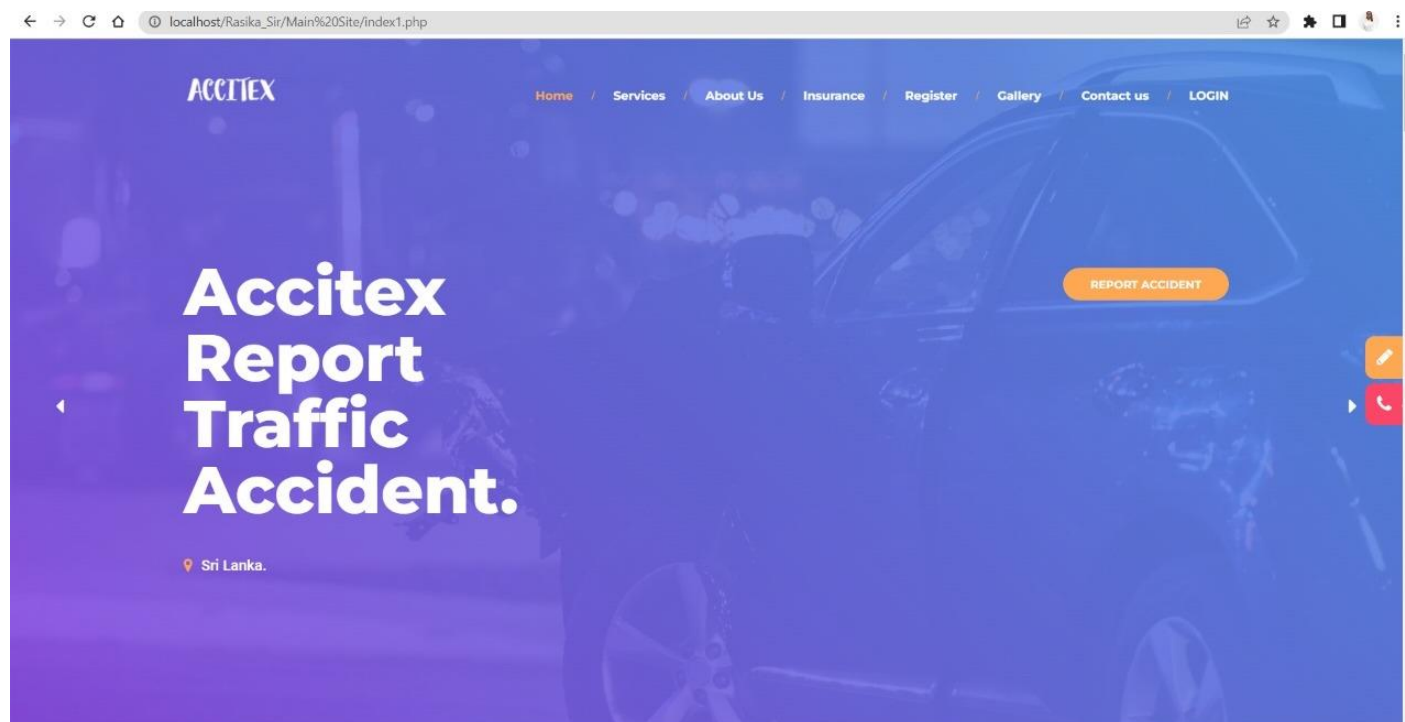
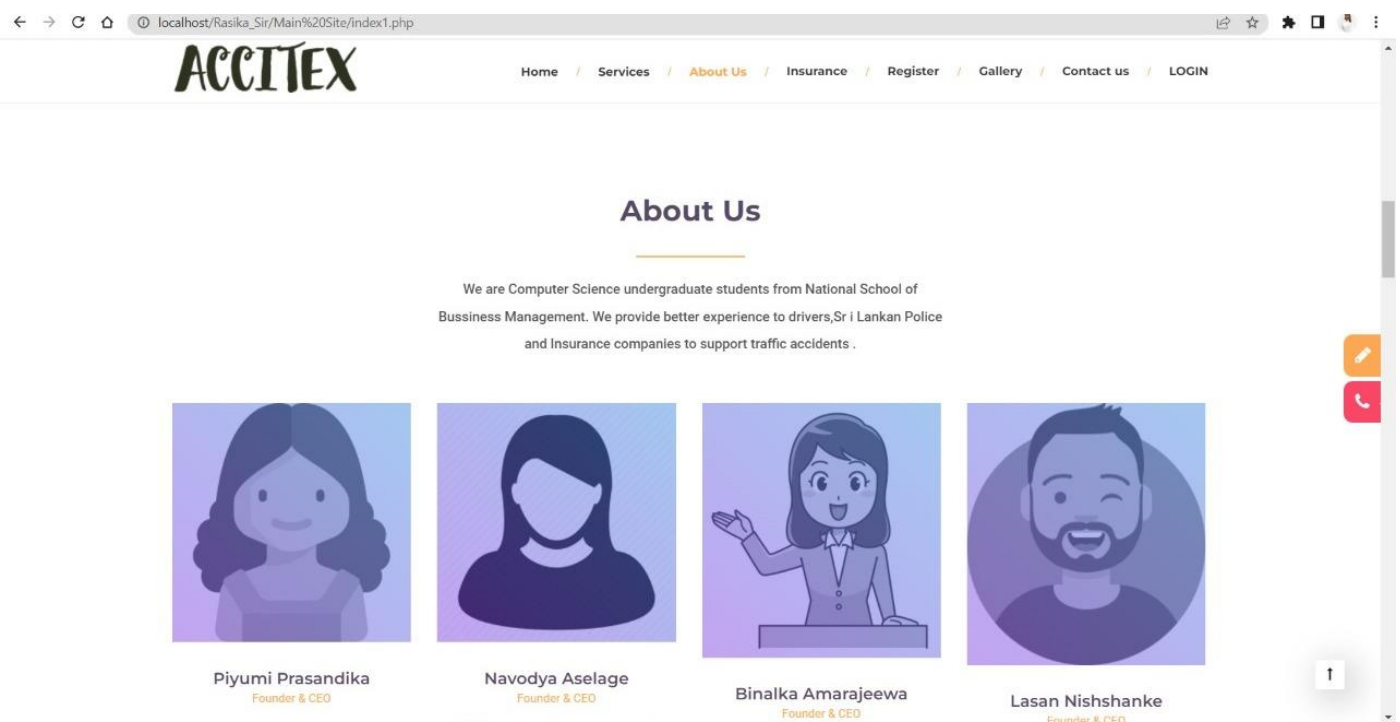


Figure 26:Home page 01



Figure 27:Home page 02



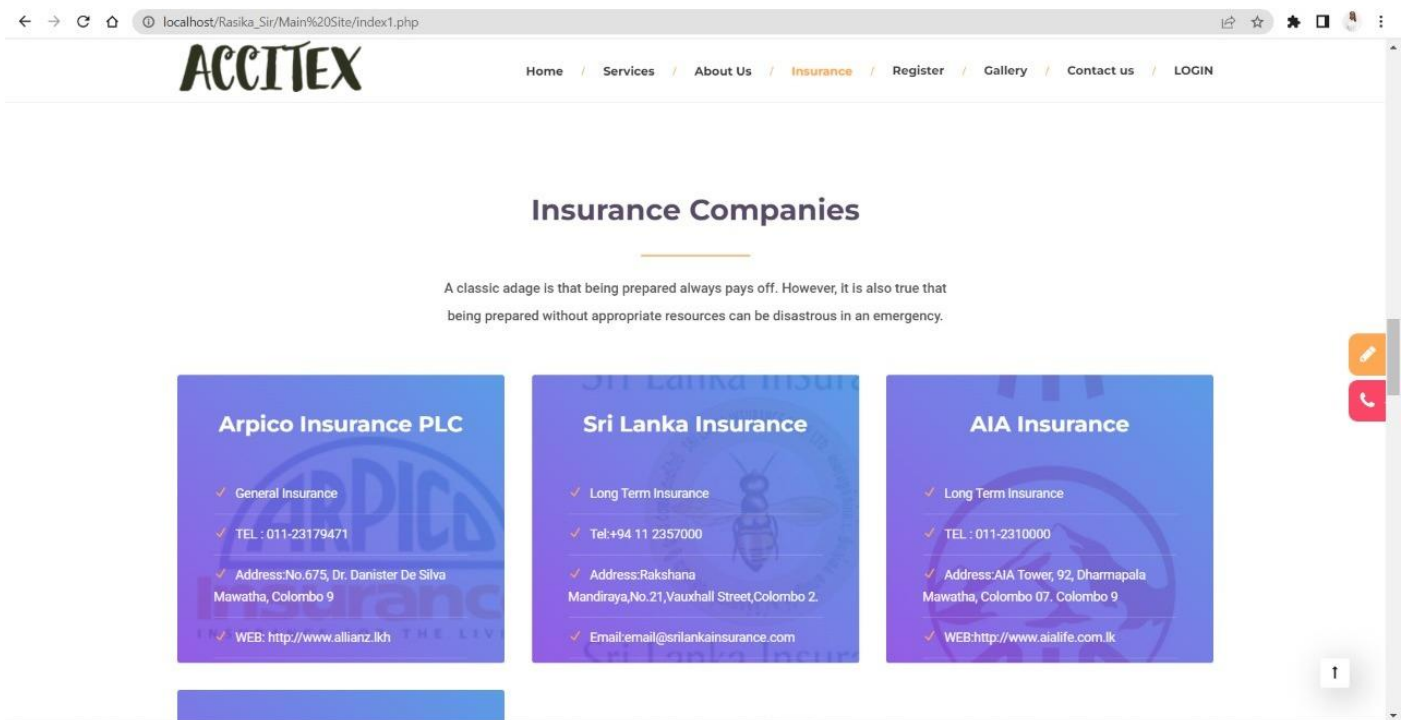


Figure 29:Main Accitex

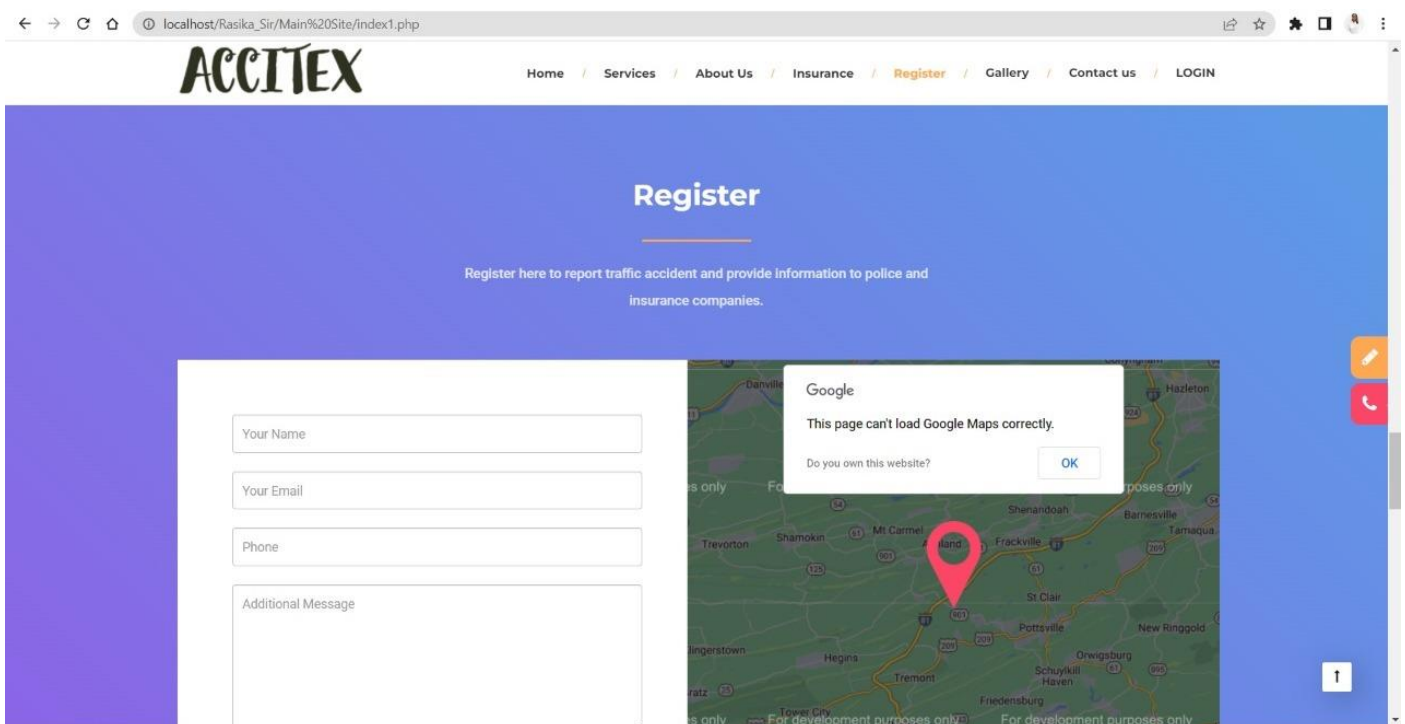


Figure 30:Register page

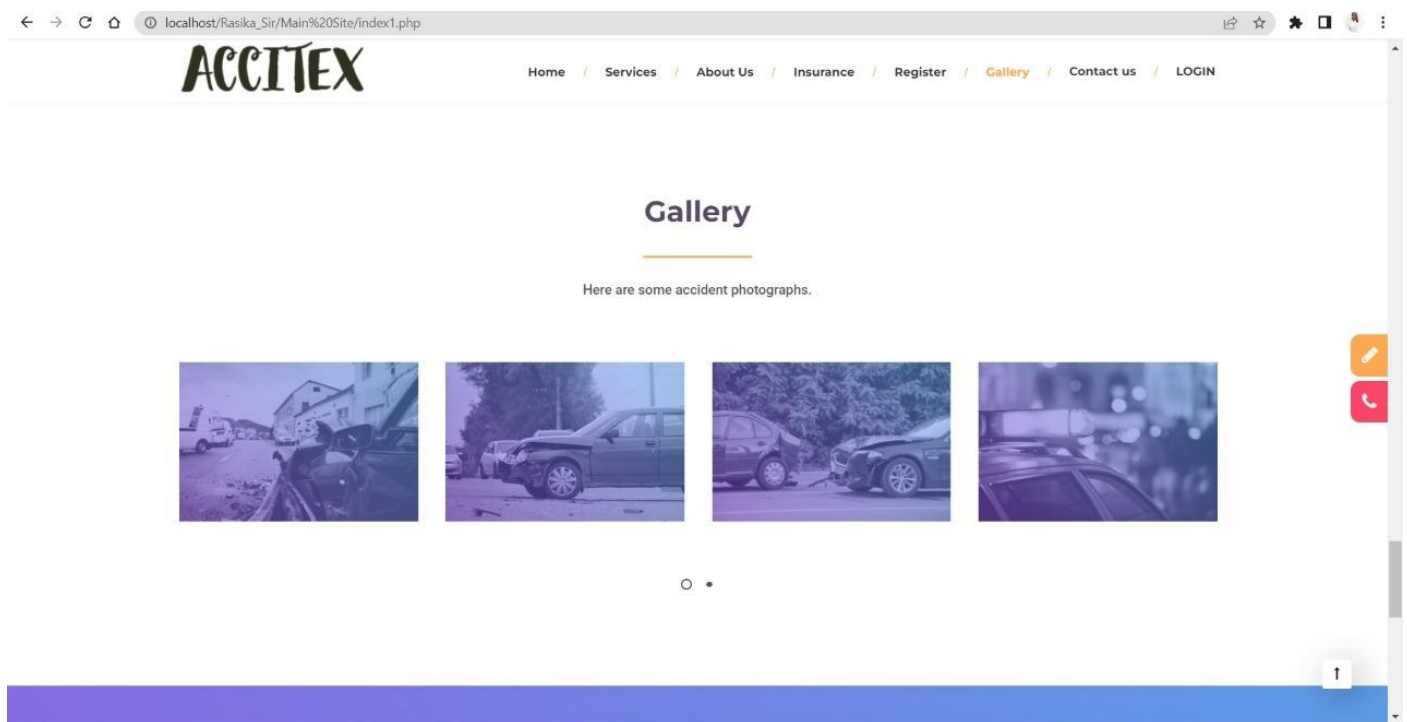


Figure 31:Gallery

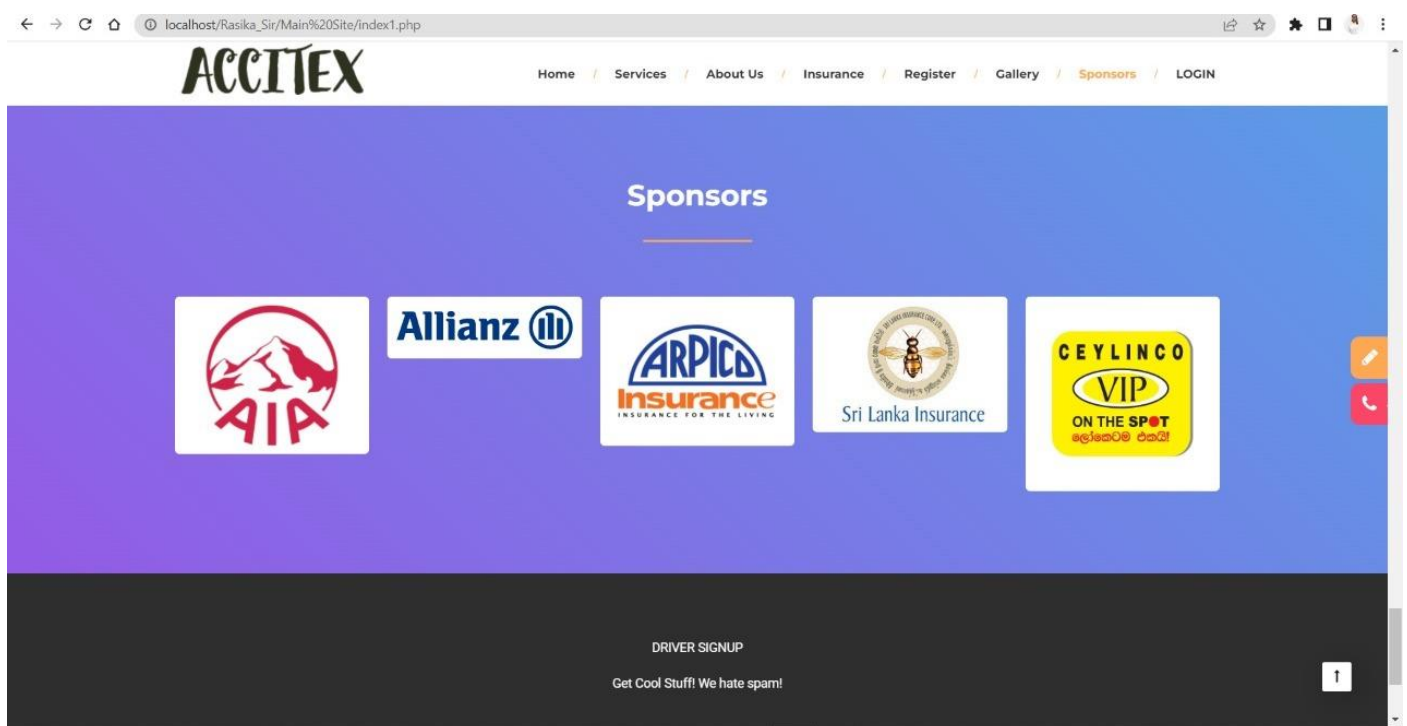


Figure 32:Sponsor

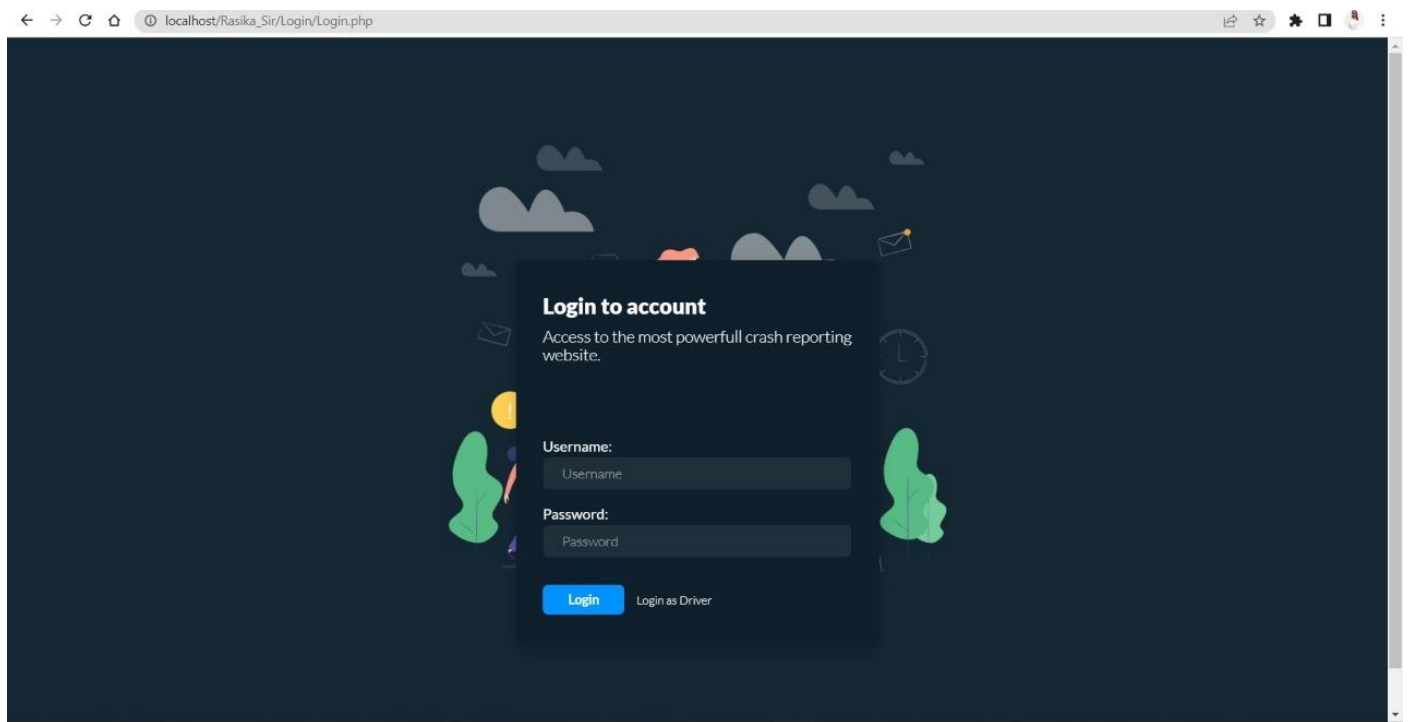


Figure 33:Login

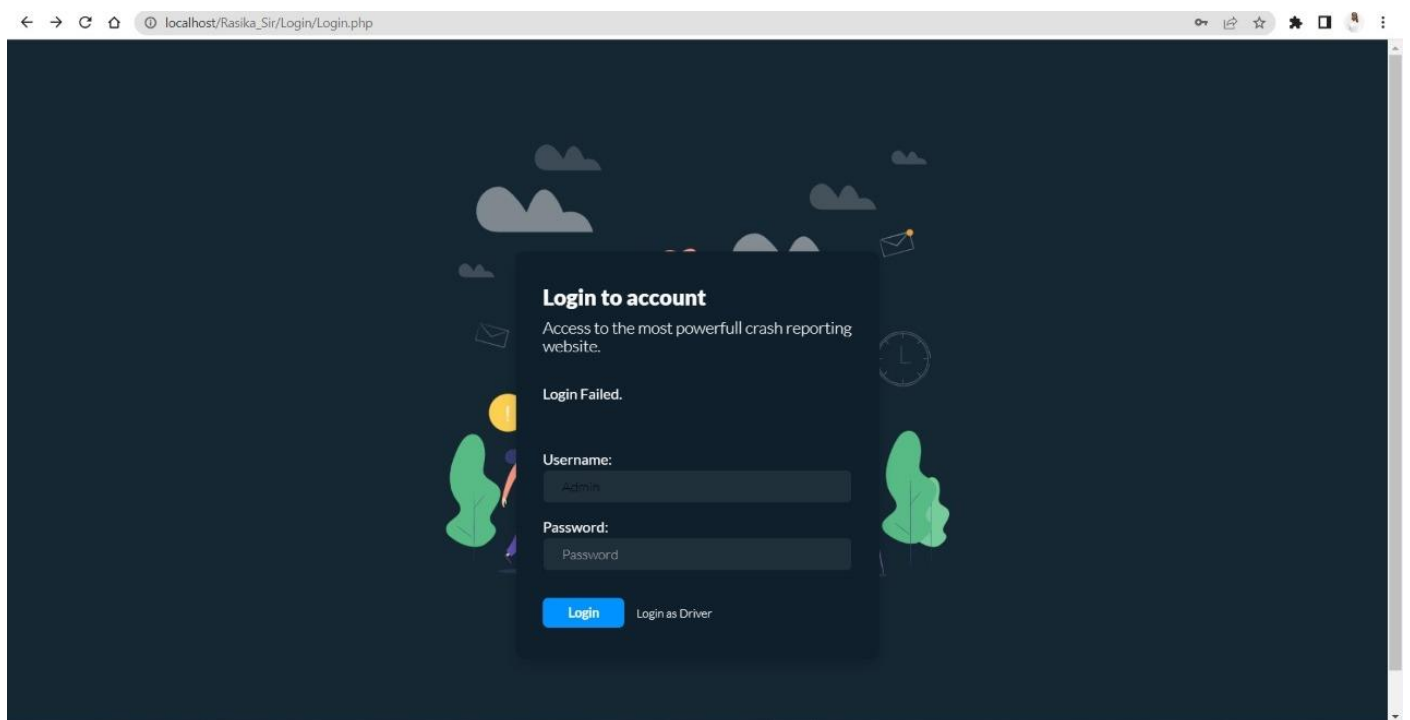


Figure 34:Login failed

The screenshot shows a web browser at the URL `localhost/Rasika_Sir/Login/admindash.php`. The page has a dark blue sidebar on the left with the 'Accitex' logo and navigation links for 'Insurance', 'Police', and 'RDA'. The main content area is titled 'Form Elements' and contains a sub-section 'Insurance Login Details'. This section includes several input fields: 'Insurance Company Name' (with value 'Company Name'), 'Insurance Company Register ID' (with value 'ASD1234'), 'Email' (with value 'insurance@example.com'), 'Company_Registered_Date' (with value 'mm/dd/yyyy' and a calendar icon), 'Telephone' (with value '1-(555)-555-5555'), 'Username' (with value 'ceylinco'), and 'Password' (with value 'ceyl23'). A purple 'Submit' button is located below the password field. On the right side of the dashboard, there is a user profile dropdown menu with options: 'Profile', 'My Wallet', 'Settings', 'Lock screen', and 'Logout'.

Figure 35:Insurance login detail

The screenshot shows a web browser at the URL `localhost/Rasika_Sir/Login/Login.php`. The page has a dark blue background with a central login form. The form is titled 'Login to account' and includes the text 'Access to the most powerfull crash reporting website.' Below this, it says 'Login Sucessful.' (note the typo). The form has two input fields: 'Username:' with the value 'Admin' and 'Password:' with the value 'Password'. A blue 'Login' button is at the bottom of the form, with a link 'Login as Driver' next to it. The background of the login page features stylized illustrations of clouds, a sun, and trees.

Figure 36:login successful

localhost/Rasika_Sir/Login/admindash.php

ACCITEX

Create ▾

Search...

English ▾

Profile

My Wallet

Settings

Lock screen

Logout

Username

Enter your username

Password

Enter your password

Submit

Sri Lankan RDA

→ Sri Lankan RDA

RDA Area

Enter RDA Area

Email

Enter your email

Contact Number

Enter your No

Username

Enter your username

Password

Enter your password

Submit

© 2022 ACCITEX - Crafted with ❤ by Group 01.

Figure 37:RDA login

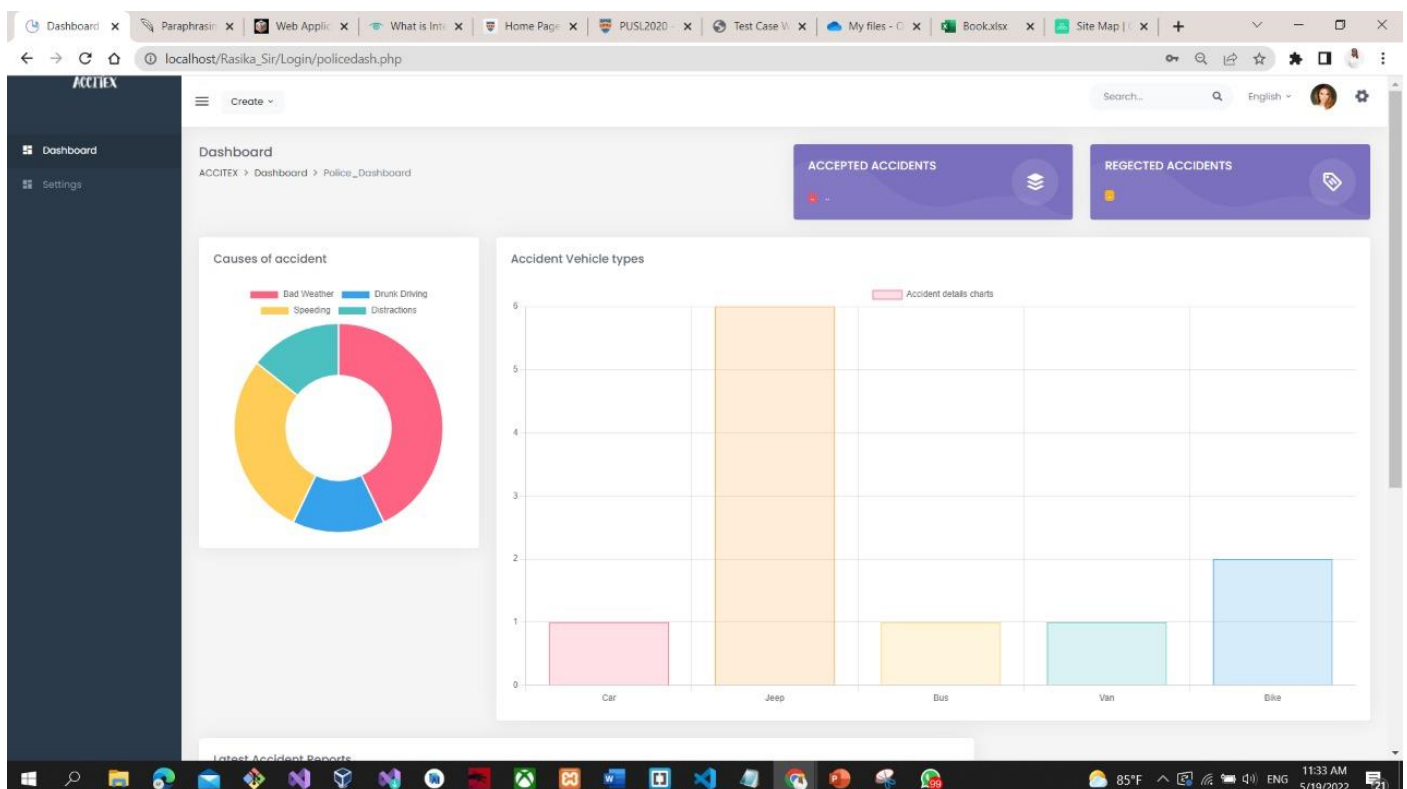


Figure 38:Charts 01

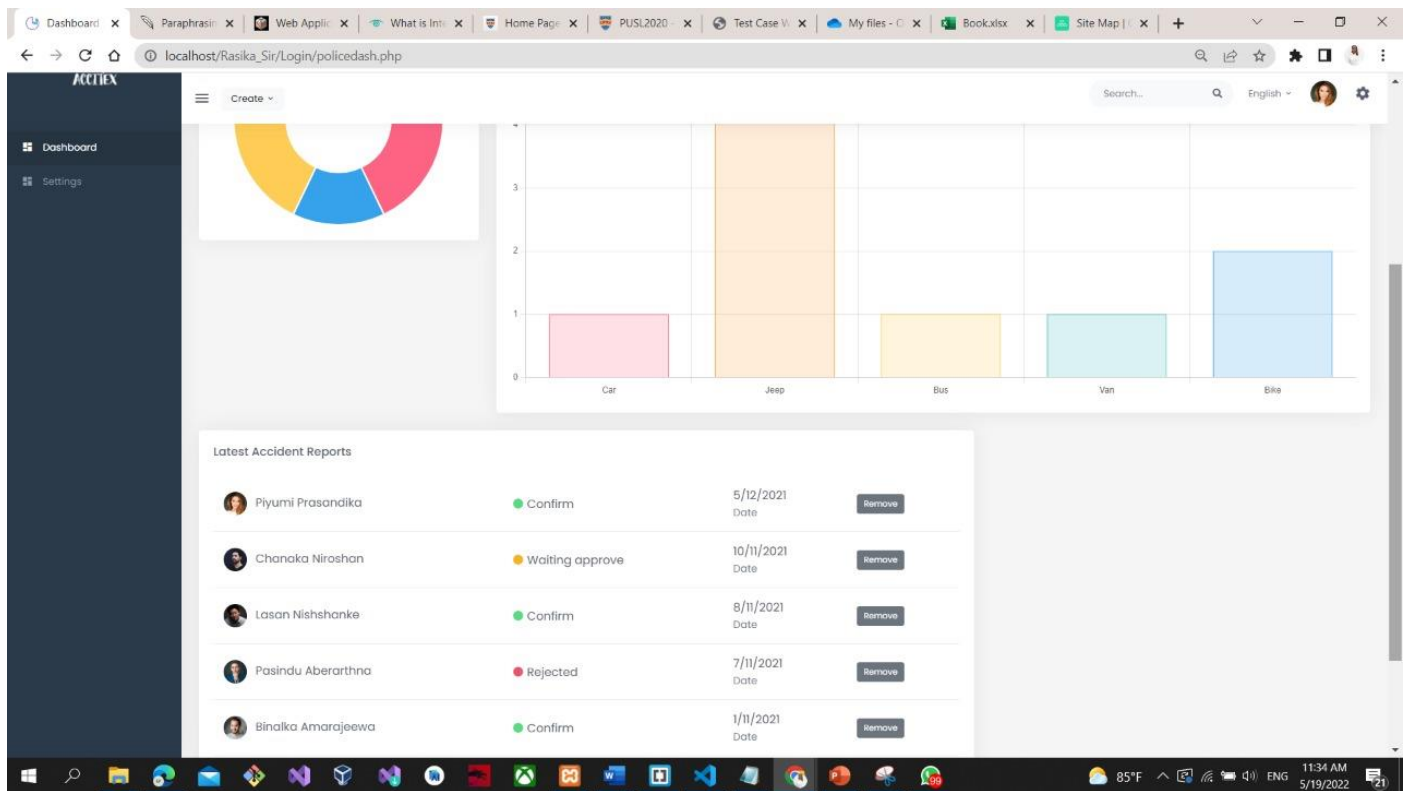


Figure 39:Charts 02

The screenshot shows the ACCITEX data table interface. The top navigation bar includes a 'Create' dropdown and a search bar. The left sidebar contains 'Dashboard', 'Police', and 'RDA' links. The main content area features a 'Data Table' section with a table of insurance data.

Insurance Company Name	Insurance Company Register ID	Email	Company_Registered_Data	Telephone	Username	Password	Action
Ceylinco	cey123	cey@gmail.com	2000-07-05	0776543432	ceylinco	cey123	Update Delete

Figure 40:Accitex data table 01

localhost/Rasika_Sir/Login/police_database.php

ACCITEX

Create

Search...

English

Dashboard

Insurance

RDA

Data Table

ACCITEX > Tables > Insurance Data Table

Police Datatable

DataTables has added Police database.

Police Area Name	Police ID	Email	Username	Password	Telephone	action
Homagama	p123	police@gmail.com	112342562	police	1234	Update Delete
Kottawa	p156	policekattawa@gmail.com	112342526	k_police	k123	Update Delete

© 2022 Accitex - Created by Group 01.

Figure 41:Accitex data table 02

localhost/Rasika_Sir/driver/driver/

Sign In To Your Account

Enter Username

Enter Password

[Don't Have An Account? Sign Up](#)

Submit

Figure 42:Driver Sign In page

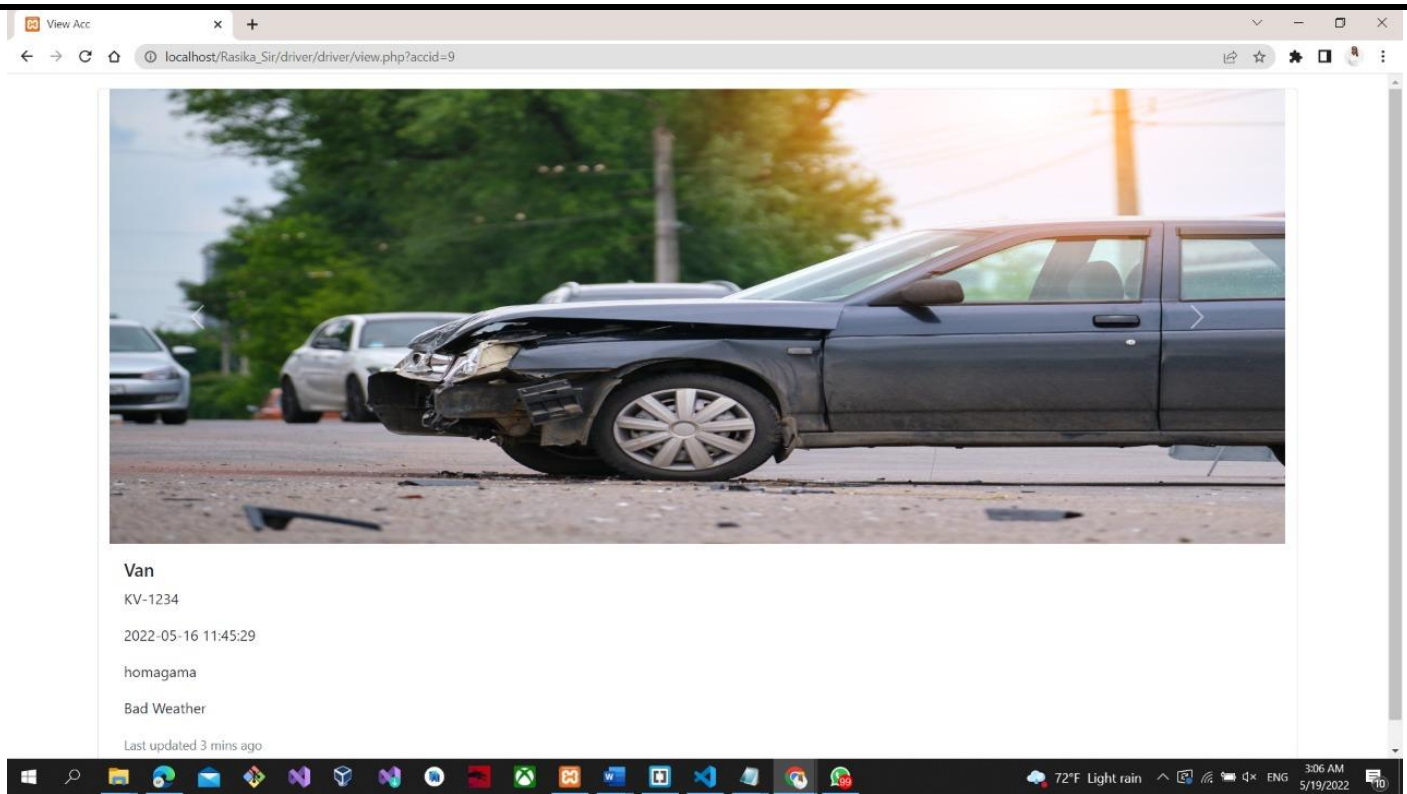


Figure 43:Accident report page

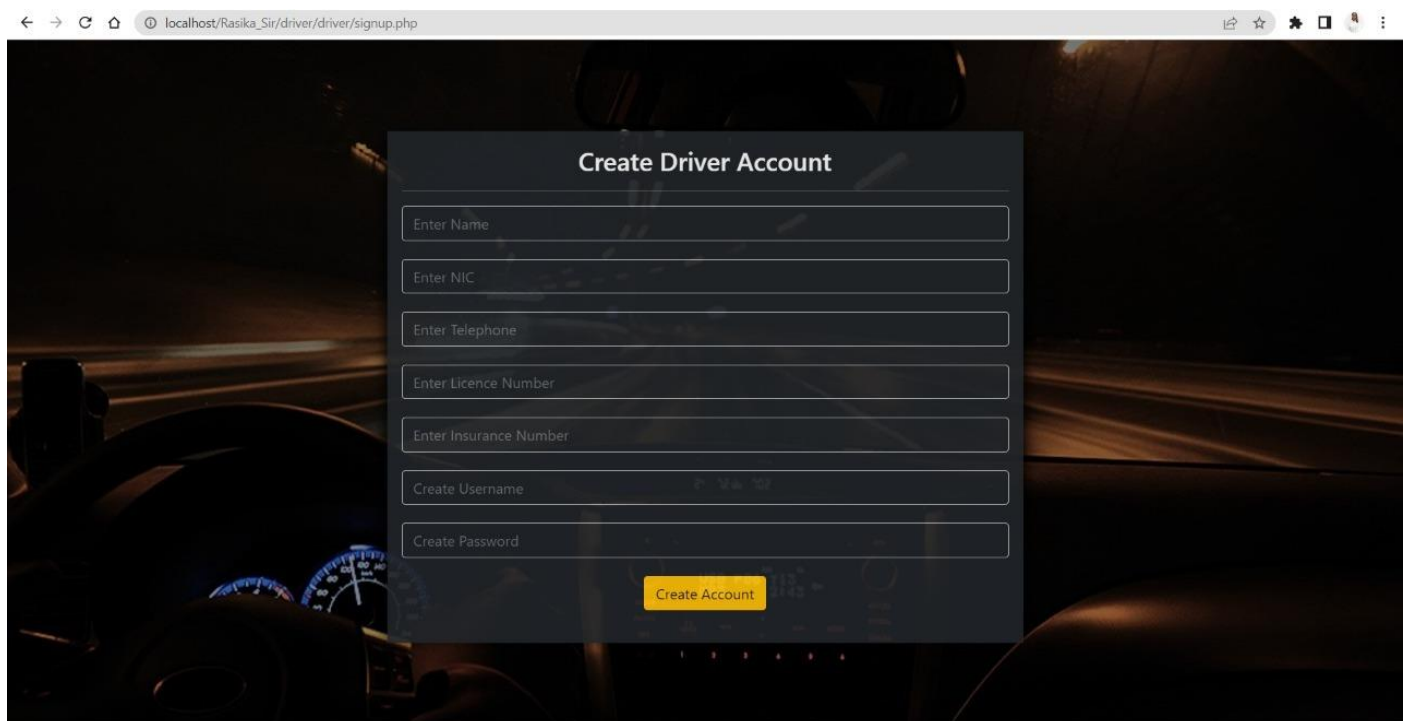


Figure 44:Create account page Driver

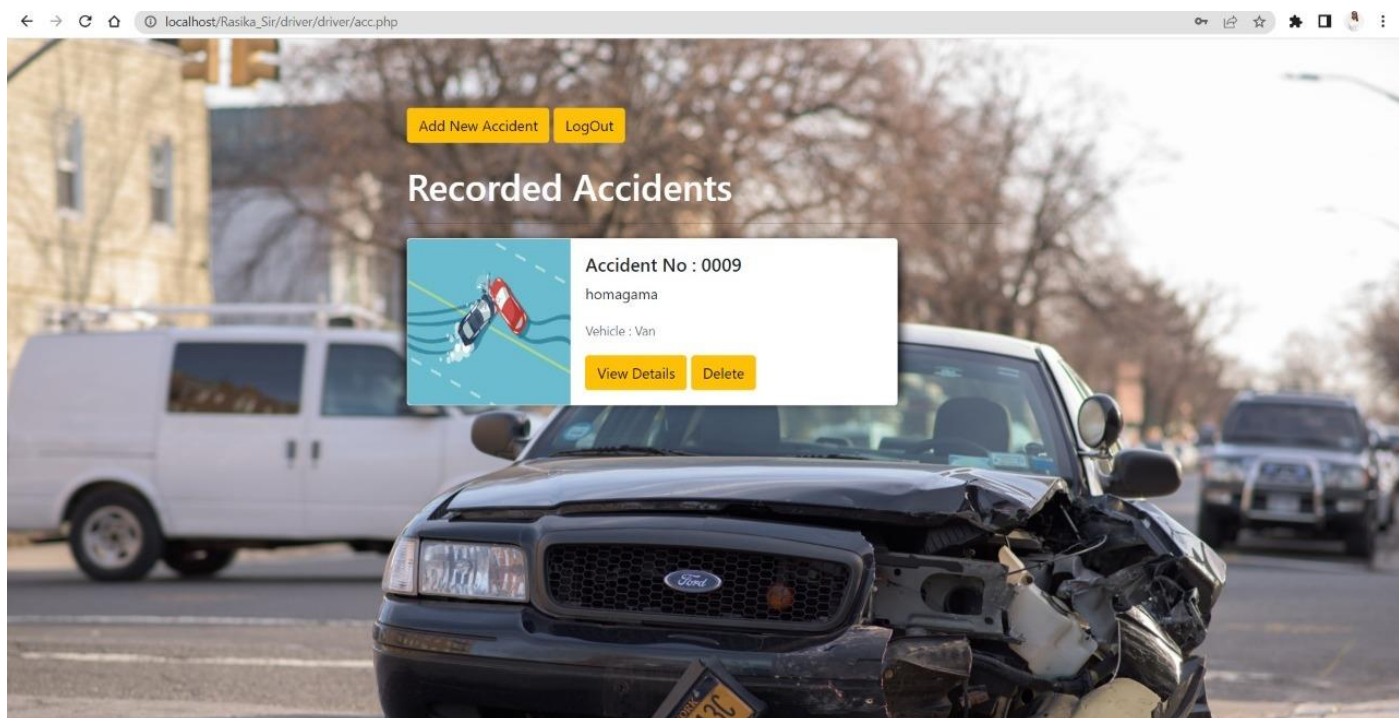


Figure 45:Recorded accident page

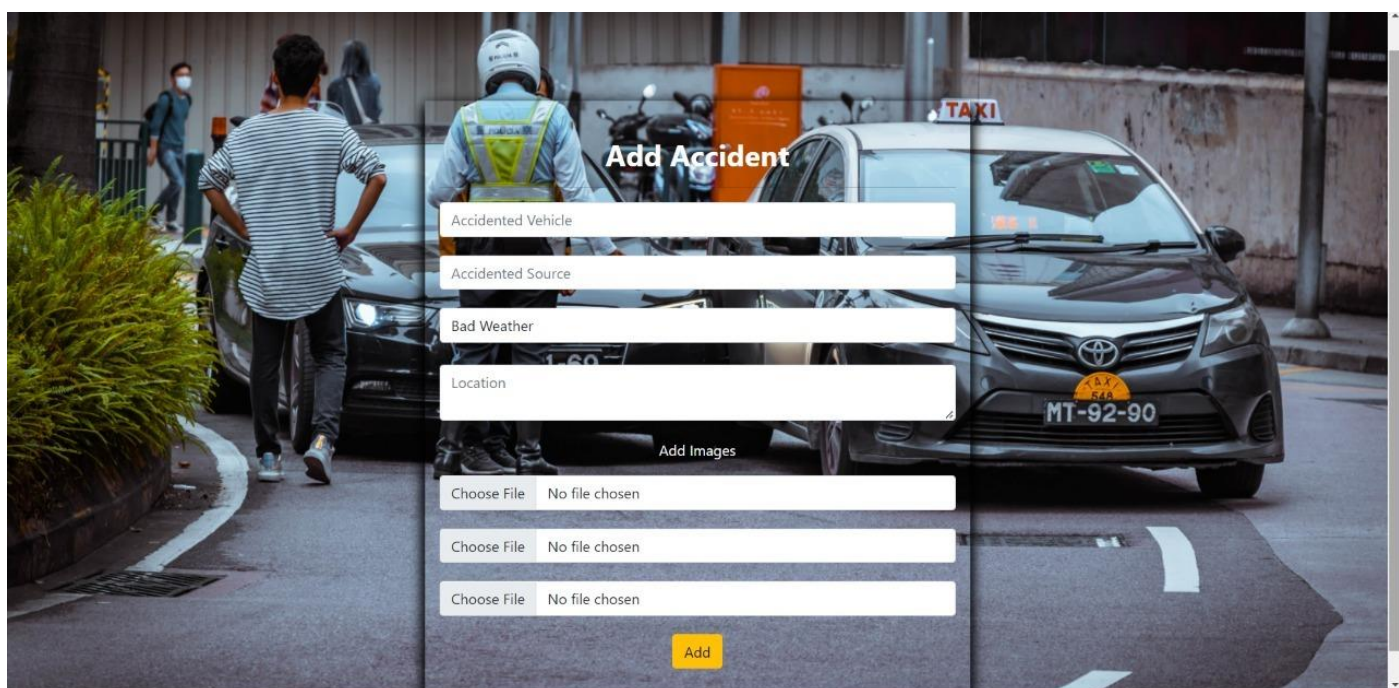


Figure 46:Add accident page

09.Team contribution

9.1 Details of members

Username	Email	Plymouth Index
Elgiriya Prasandika	10749850@students.plymouth.ac.uk	10749850
Swarnathilaka Amarajeewa	10749949@students.plymouth.ac.uk	10749949
Asalage Navodya	10749851 @students.plymouth.ac.uk	10749851
Nishshankage Nishshanka	10749846@students.plymouth.ac.uk	10749846
Egallalage Nandasiri	10749855@students.plymouth.ac.uk	10749855
Mahadurage Abeyrathna	10749852 @students.plymouth.ac.uk	10749852

9.2 Contribution Statement of group members.

1. Elgiriya Prasandika

As the team leader for this project, I divided the workload between the members of our group.

In this effort, the website will provide a platform for drivers to report traffic accidents and accidents. Our website is titled "ACCITEX." The webmaster will build accounts for Insurance Companies, Sri Lankan Police, and the RDA on this website. Administrators and insurance firms have the same login page. I created this login page using the languages bootstrap, CSS, HTML, JS Query, PHP, etc., as well as an SVG background image. I validate the login page using SESSIONS in this section. As a result of integrating three tables into a single login, however, creating this page was fraught with difficulties. As the Web Master of our website, I designed three forms to collect insurance, police, and RDA information for account creation. On the login page, the login-check PHP code was used to test whether the username and password match the added information.

Created the website's homepage using bootstrap, HTML, JS, CSS, and jQuery, and incorporated accident-related images.

2. Swarnathilaka Amarajeewa

In this project I have contributed for the Drivers Sign in and Sign-Up pages.

login: -

Designing the login form with html css with Bootstrap. when user already have an account user can login to the system if not user can create an account by clicking the signup option. In here I have used SQL data retrieve query to login process. By entering correct username and password use can log to the system. If the logging is successful, the system pop-up a message as "login successful" if not system display "login unsuccessful" message. Here the messages are getting by validation process.

signup: -

This signup form, I will use Html css and bootstrap for designing the interface. user can fill all these fields for create an account. By clicking the create account button, entered all data store into the database using SQL insert query with PHP. After that user redirect to the login page and user can log to the system

3. Asalage Navodya

I have contribute for the **web master**. Here the web master is creating a username and a password. For this. I have used PHP, Html, CSS, Bootstrap, JS. All the inputting details are stored inside this table. There are 03 forms in this application 01 is for police officers.02 is for insurance officers and the last is for RDA officers. RDA and insurance data is stored under a one table. There are 2 tables which contain data. 01 table store the details of police officers and the second table stores the details of insurance officers. The inputting data can be deleted and update by the buttons provided in relevant pages.

4. Nishshankage Nishshanka

In this project, I have contributed for making the **police information page**. The police can see the accidents approved by RDA staff. These data are shown by a graph. I did this part of the project with MySQL and a JavaScript chart library called "Chart JS". First, the accident data from MySQL tables are getting to the variables in PHP as values. Next, I implemented those values using Java Script to the Chart JS library's data array. Then police can see the details about accidents. But there is no auto-refresh feature in these charts. If a user logged into the site and adds an accident, the details are shown in the charts after refreshing the police's page. The chart includes details such as percentages of the accidents according to the vehicles, causes for the accidents, and fluctuations of accidents now and then. All the details are getting from the MySQL database and shown in HTML using PHP variables. At the designing part of the police's page, I've not used any custom styling for charts, and the default one is used. The charts are mobile responsive. But some parts of the page are not mobile responsive. The PHP and JavaScript codes are not well optimized but work perfectly in any modern web browser without any issue. When I'm wrote code for charts, I faced some issues. One of them is some data is not shown in the chart. I resolved that issue with the web browser's (I used google chrome) JavaScript console using the 'console.log ()' function. I know it is not a perfect method for that, but I was able to get some results from that function. For the testing part, I used a method called Rubber Duck Debugging. Finally, I checked PHP, JavaScript codes, and MySQL queries A to Z to confirm whether the Queries, Charts, and the result are working well

5. Egallalage Nandasiri

Here I Contribute for the **Insurance detail page**. I did this to display the accident details given by a driver in the common dashboard. I used HTML, CSS, Bootstrap. Insurance officers can accept or reject accident claims done by the drivers. Then accepted accidents goes to approved accident details and rejected accidents remove from the dashboard in this process. The images uploaded by user can vies from the dashboard.

6. Mahadurage Abeyrathna

. Here, I have contributed for the **Add accident and View accident pages.**

Add accident: -

I have use html css and bootstrap for the add accident page. To get an interactive user interface in here, all user of the system can add an accident through filling this add accident form. The driver can add to the database by filling in the accident details form. we can add some details of the accident and some photos of the relevant accident by clicking the add button, all filled data and pictures are saved into the database for the storing process I use SQL insert query using PHP

view accident: -

In here I build a connection between database and view page using PHP. What happens here is that the data about the accident entered by a driver is viewed through this. Also, what is specifically included here is that a driver can update or delete the data they have entered. I am using SQL retrieve query to retrieve the data from the database in this view page we can see details of related accident withing images