

Index of Contents

1. Description of Functional Requirements

- a. Admin Module
 - i. Login
 - ii. Add/Delete/Edit Routes
 - iii. Add/Delete/Edit Announcements
 - iv. Add/Delete/Modify/View User Subscriptions
 - v. View Current Position of the shuttles
- b. User Module
 - i. Sign up/Login
 - ii. Driver can send update current position
 - iii. Driver can send an announcement
 - iv. Other users can view current position of the shuttle
 - v. Users can also view the waiting time in any stoppage
- c. Driver Module
 - i. Add/Delete Announcements

2. Description of Interface Requirements

- a. Allocation type
- b. Front-end software
- c. Back-end software
- d. Operating System
- e. Supported Browsers
- f. Editors and Compiler Used

3. Description of Nonfunctional Requirements

- a. Testing
 - i. Field Entry Validation
 - ii. Output Verification
 - iii. Security
- b. Roles of the Team Members
 - i. Project Manager
 - ii. Front-end Developers
 - iii. Back-end Developers
 - iv. Quality Assurance Analysts
 - v. Documentation Team
- c. Project Experience

Richie Molina, Mishell Quispe,
Danny Salas, Renee Rodas,
Victor Lliguicota

Dr. Kazi Zakia Sultana
CSIT 415-03 SP23
05/09/23

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

The screenshot shows a web browser window with multiple tabs open at the top, including 'localhost:8000/registration.php'. The main content area displays the 'University Shuttle Bus App' registration page. The page has a dark background with white text. At the top center is a large 'Welcome to the University Shuttle Bus App' heading, flanked by a red alarm clock icon on the left and a blue bus icon on the right. Below the heading is a sub-headline: 'Get around campus with ease. Find the nearest shuttle bus stop, get real-time schedules, and track your bus in real-time.' followed by 'More updates to functionality coming soon. Please check back again soon.' with two small emoji faces. The central part of the page is a 'Create a new account' form. It includes fields for 'User name' (with placeholder 'user name'), 'Email' (placeholder 'email address'), and 'Password' (placeholder 'password'). Below these is a 'User Type' section with a dropdown menu titled 'Select User Type' containing options: Admin, User, Driver, Guest, Visitor, Faculty, Staff, and Student. At the bottom of the page, a footer bar contains links for 'Privacy Policy', 'Terms of Service', and the copyright notice '© 2023 University Shuttle Bus App. All Rights Reserved.'

Richie Molina, Mishell Quispe,
Danny Salas, Renee Rodas,
Victor Lliguicota

Dr. Kazi Zakia Sultana
CSIT 415-03 SP23
05/09/23

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

The screenshot shows a web browser window with multiple tabs open at the top, including 'Subscriptions - YouTube', 'How to Create a Chat', 'Instance details | EC2', 'University Shuttle Bus', 'Inbox (12) - jrmolin90', 'Inbox (677) - molinar6', 'CSIT340 Project Essays', and 'University Shuttle Bus'. The main content area displays the 'University Shuttle Bus App' login page. The page has a dark background with a central light gray box. At the top of this box is a red alarm clock icon and the text 'Welcome to the University Shuttle Bus App' followed by a blue bus icon. Below this, there is a message: 'Get around campus with ease. Find the nearest shuttle bus stop, get real-time schedules, and track your bus in real-time.' and 'More updates to functionality coming soon. Please check back again soon.' with two small emoji faces. At the bottom of the light gray box is the word 'Log in' in white. Below 'Log in' are two input fields: 'Email' with placeholder 'Enter email' and 'Password' with placeholder 'Password'. Underneath these fields are two buttons: a blue 'Login' button and a green 'Register' button. At the very bottom of the page, in a small white font, is the text 'Privacy Policy · Terms of Service © 2023 University Shuttle Bus App. All Rights Reserved.'

Software Engineering II: Project Report
 Group 6: University Shuttle Bus App

Instance summary for i-0cd7a2359bafe888a (lamp bus_app)

Networking

- Public IPv4 address: 34.194.204.120
- Private IP4 DNS: ec2-34-194-204-120.compute-1.amazonaws.com
- Elastic IP addresses: 34.194.204.120 [Public IP]

Storage

- AMI ID: ami-0557a15b87f6559cf
- AMI name: ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230208
- Kernel ID: -
- RAM disk ID: -

Monitoring

- Monitoring: disabled
- Termination protection: Disabled
- AMI location: amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230208
- Stop-hibernate behavior: disabled
- State transition reason: -
- State transition message: -
- Owner: 243400111554
- Boot mode: -
- Use RBN as guest OS hostname: -

Tags

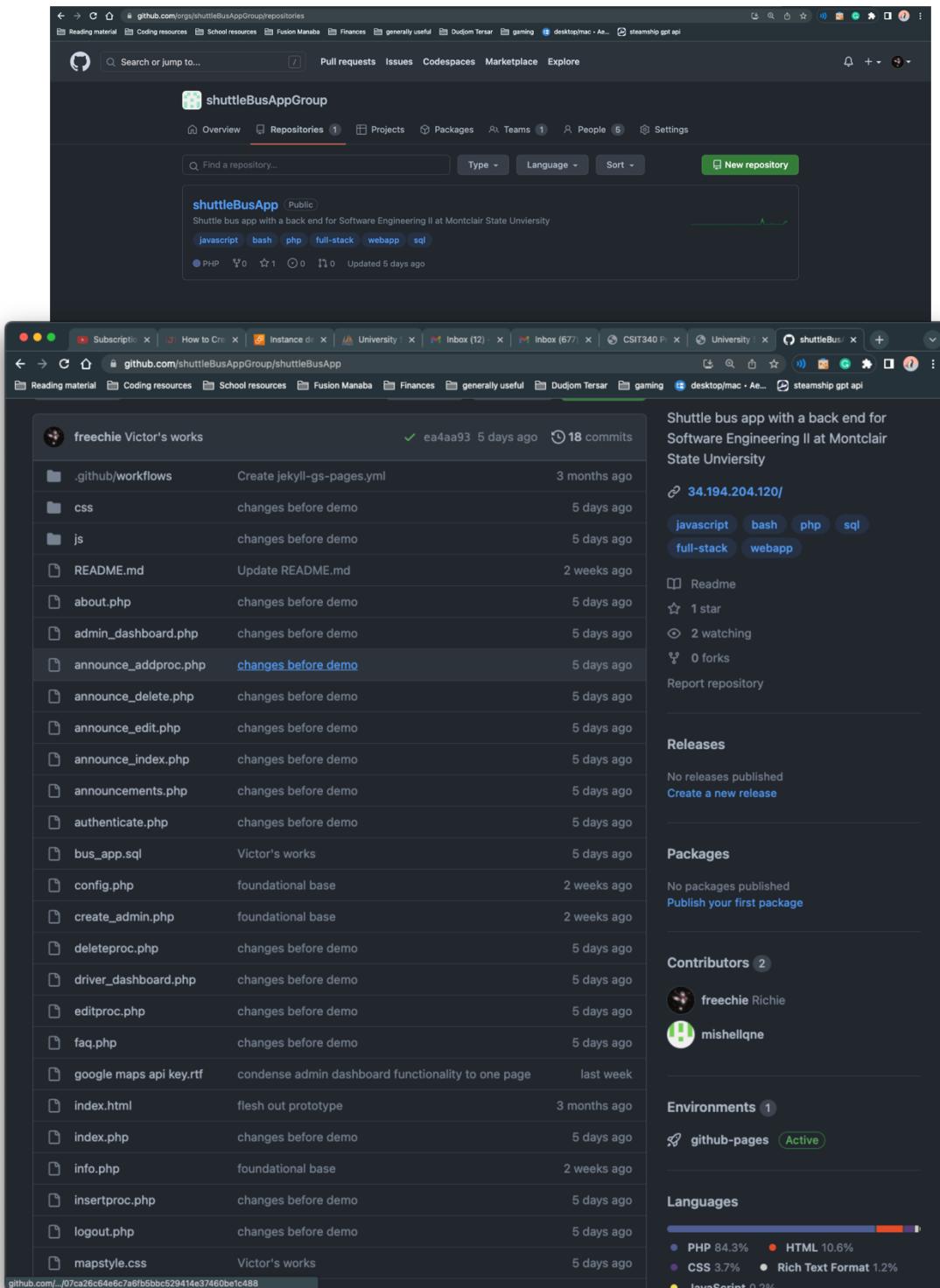
Key	Value
lamp bus_app	i-0cd7a2359bafe888a

Richie Molina, Mishell Quispe,
Danny Salas, Renee Rodas,
Victor Lliguicota

Dr. Kazi Zakia Sultana
CSIT 415-03 SP23
05/09/23

Software Engineering II: Project Report

Group 6: University Shuttle Bus App



<https://github.com/shuttleBusAppGroup/shuttleBusApp>

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

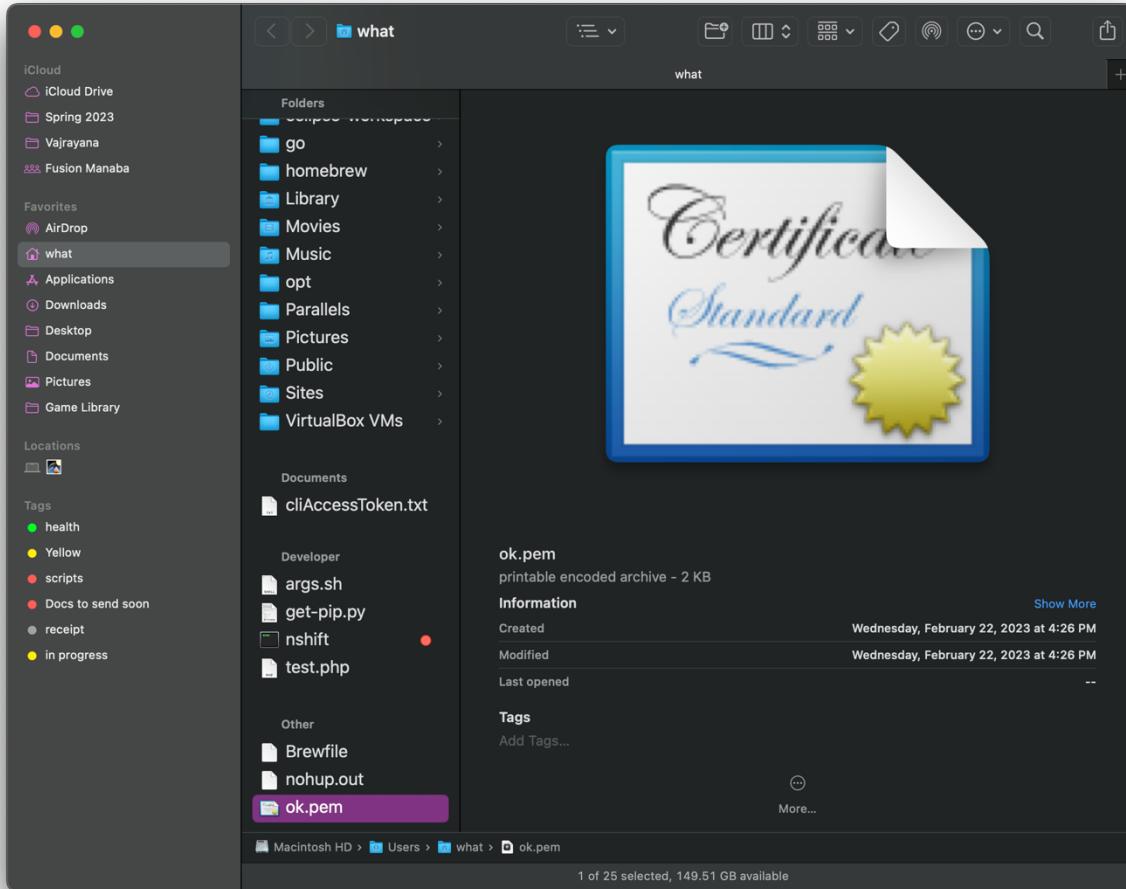


Figure 1 AWS EC2 Instance security Key used in ssh below

Software Engineering II: Project Report
 Group 6: University Shuttle Bus App

The left terminal window shows the output of a system status command:

```
[~] ssh -i "ok.pem" ubuntu@ec2-34-194-204-120.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1024-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Wed May 10 21:29:54 UTC 2023

 System load: 0.013671875 Processes: 131
 Usage of /: 17.9% of 28.89GB Users logged in: 0
 Memory usage: 27% IPv4 address for eth0: 172.31.57.227
 Swap usage: 0%

 * Ubuntu Pro delivers the most comprehensive open source security and
 compliance features.

 https://ubuntu.com/aws/pro

 Expanded Security Maintenance for Applications is not enabled.

 1 update can be applied immediately.
 To see these additional updates run: apt list --upgradable

 5 additional security updates can be applied with ESM Apps.
 Learn more about enabling ESM Apps service at https://ubuntu.com/esm

 Last login: Fri May  5 19:22:40 2023 from 130.68.183.14
ubuntu@ip-172-31-57-227:~$
```

The right terminal window shows the output of a MySQL query to list all tables:

```
....com - ttys001 ... - ttys000
1 row in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_bus_app |
+-----+
| announcements
| buses
| driver_bus_assignments
| driver_shifts
| drivers
| route_stops
| routes
| schedules
| stops
| subscriptions
| user_subscriptions
| users
+-----+
12 rows in set (0.00 sec)

mysql>
```

Figure 2 All tables in production for shuttle bus app

The terminal window shows the contents of the `/var/www/html` directory:

```
[ubuntu@ip-172-31-57-227:/var/www/html$ ls
about.php           deleteproc.php      registration.php
addproc.php         driver_dashboard.php retrieve.php
admin_dashboard.php editproc.php       scheduleindex.html
announce_addproc.php faq.php          shuttle_position.php
announce_delete.php index.php         subscriptions.php
announce_edit.php   info.php          terms-of-service.php
announce_index.php insertproc.php    test.php
announcements.php  js                timetracking.html
authenticate.php   mq_user_dashboard.php update_current_stop.php
config.php          logout.php        user_dashboard.php
create_admin.php   privacy-policy.php
css                register.php
```

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

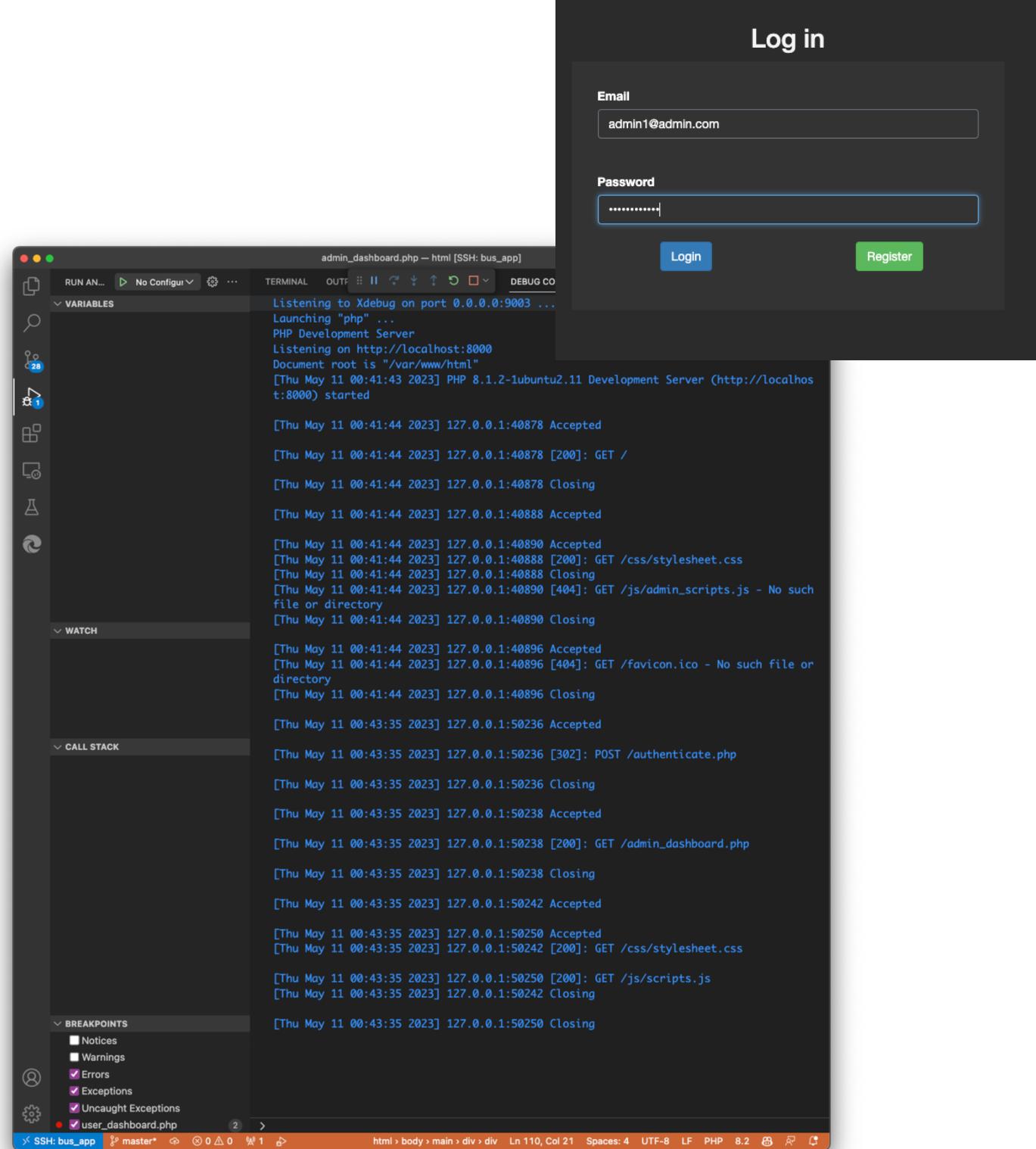


Figure 3 authenticating user and redirecting based off user type after clicking Login button

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

The screenshot shows a web-based admin dashboard for a university shuttle bus system. At the top, there's a navigation bar with links like 'Admin Dashboard', 'Home', 'About', 'FAQs', and 'Log out'. Below this is a large central area divided into two main sections: 'Insert a New Route' and 'Update or Delete Routes'.

Insert a New Route: This section contains several input fields with validation messages:

- Route ID:** A text input field with the placeholder "Please enter a route ID."
- Route Name:** A text input field with the placeholder "Please enter a route name."
- Origin:** A text input field with the placeholder "Please enter an origin for this route."
- Destination:** A text input field with the placeholder "Please enter a destination for this route."
- Bus Number:** A text input field with the placeholder "Please enter a bus number for this route."
- Arrival time:** A dropdown menu with the placeholder "Please enter an arrival time." containing options like "00:30:00", "01:00:00", and "01:15:00".

A blue "Submit" button is located at the bottom left of this section.

Update or Delete Routes: This section displays a table of existing routes:

Route ID	Route Name	Origin	Destination	Stops	Estimated Time	Bus Number	Action
1	Red Route	Main Campus	Downtown Campus	Stop 1, Stop 2, Stop 3	00:30:00	1	<button>Edit</button> <button>Delete</button>
2	Blue Route	North Campus	South Campus	Stop 1, Stop 2, Stop 3, Stop 4	01:00:00	2	<button>Edit</button> <button>Delete</button>
3	Green Route	East Campus	West Campus	Stop 1, Stop 2, Stop 3, Stop 4	01:15:00	3	<button>Edit</button> <button>Delete</button>

At the bottom of the page, there's a footer with links to "Privacy Policy" and "Terms of Service", followed by the copyright notice "© 2023 University Shuttle Bus App. All Rights Reserved."

Figure 4 Admin Dashboard

Description of Functional Requirements

Admin Module

To log in, we have the following php security measures set in place to restrict content on this website according to current user session. The user is validated upon hitting the login button via secure PHP scripts. Then the validated user is redirected to the specific dashboard given that their current logged in session describes the current user as its pertinent user_type depending on the dashboard. In this case, we have a clause set for admin. You will see the same behavior for the user and driver.

```
10 // Check if the user is logged in and has the correct user_type
11 if (!isset($_SESSION["user_id"]) || $_SESSION["user_type"] != "admin") {
12     header("Location: index.php");
13     exit();
14 }
```

At the production level, the admin dashboard features capabilities to manipulate routes, announcements, and user subscriptions. The admin can also see where the buses on all routes are currently located. Below we see a screenshot of our dashboard which has incorporated most of these features in one page. We are working on unifying all admin features into a single dashboard page. Mishell's section further into this report will demonstrate the Add/Delete/Edit Routes & Add/Delete/Edit Announcements

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

```
| user_subscriptions | CREATE TABLE `user_subscriptions` (
|   `id` int NOT NULL AUTO_INCREMENT,
|   `user_id` int NOT NULL,
|   `subscription_id` int NOT NULL,
|   `start_date` date NOT NULL,
|   `expiry_date` date NOT NULL,
| PRIMARY KEY (`id`),
| KEY `user_id` (`user_id`),
| KEY `subscription_id` (`subscription_id`),
| CONSTRAINT `user_subscriptions_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
| ) ON DELETE CASCADE,
| CONSTRAINT `user_subscriptions_ibfk_2` FOREIGN KEY (`subscription_id`) REFERENCES `subscriptions` (`id`) ON DELETE CASCADE
| ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

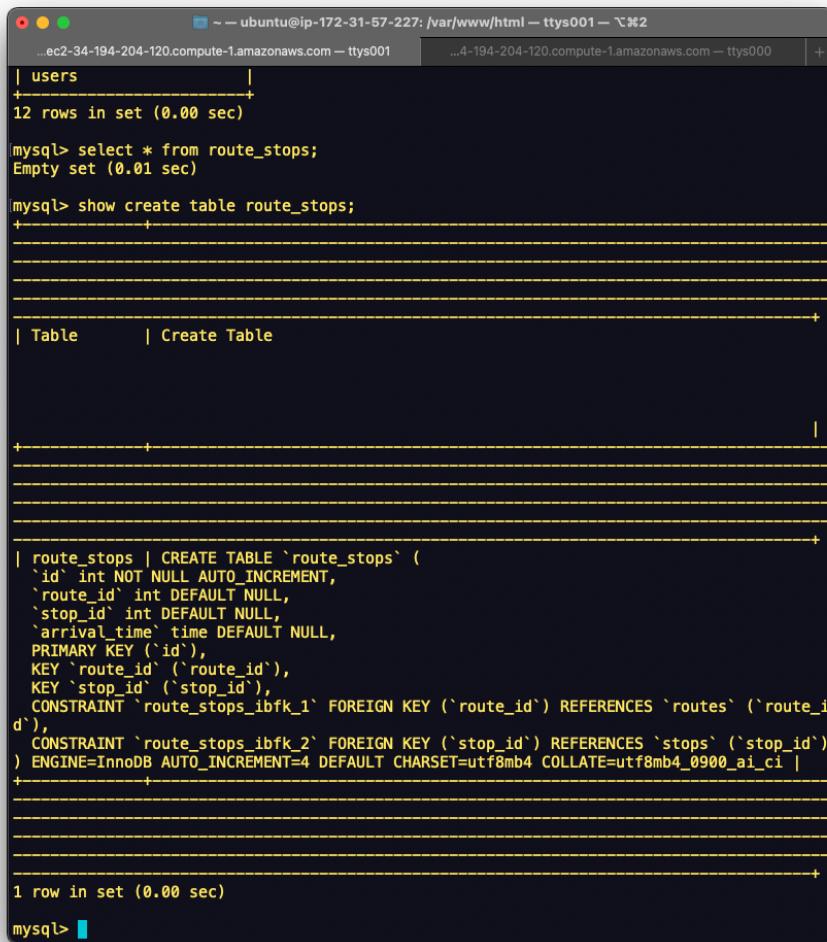
mysql> select * from user_subscriptions;
+----+-----+-----+-----+-----+
| id | user_id | subscription_id | start_date | expiry_date |
+----+-----+-----+-----+-----+
| 1 | 9 | 8 | 2023-05-03 | 2023-06-02 |
| 2 | 11 | 10 | 2023-05-03 | 2023-06-02 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 5 user subscriptions table

The table above shows the contents of the MySQL table containing data in the ‘user_subscriptions’ table. We are working to have the feature implemented into the admin dashboard where the admin can modify this on a per user basis.

Software Engineering II: Project Report
Group 6: University Shuttle Bus App



The screenshot shows a terminal window with two tabs. The left tab displays the output of a 'users' query, showing 12 rows in set (0.00 sec). The right tab displays the output of a 'route_stops' query, showing an empty set (0.01 sec). Below these, the 'show create table route_stops;' command is run, displaying the table's definition. The table 'route_stops' is created with the following schema:

```
CREATE TABLE `route_stops` (
  `id` int NOT NULL AUTO_INCREMENT,
  `route_id` int DEFAULT NULL,
  `stop_id` int DEFAULT NULL,
  `arrival_time` time DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `route_id` (`route_id`),
  KEY `stop_id` (`stop_id`),
  CONSTRAINT `route_stops_ibfk_1` FOREIGN KEY (`route_id`) REFERENCES `routes` (`route_id`),
  CONSTRAINT `route_stops_ibfk_2` FOREIGN KEY (`stop_id`) REFERENCES `stops` (`stop_id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

1 row in set (0.00 sec)

mysql>

Figure 6 Route Stops Table

The table above stores the route stop information. On the announcements, the driver will have the capability to announce his next stop. This data store will store the logic that changes the values by reference as “next stop” is called in the announcement text field.

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

User Module

The screenshot shows a web browser window with a dark theme. The address bar displays "localhost:8000/user_dashboard.php". The page header includes links for "User Dashboard", "Home", "About", "FAQs", and "Log out".

User Dashboard

Welcome, testuser!
Current East Coast Time: 09:28 PM

Shuttle Bus Announcements

Title	Message	Added On
Bus Maintenance	Bus 3 will be under maintenance on May 12th.	2023-05-05 12:00:00
testing	hello	2023-05-05 14:26:50
testerter	test	2023-05-05 14:34:57
New Route Added	We have added a new Green Route for easier travel between campuses.	2023-05-05 15:00:00
testing again	what in the	2023-05-05 14:42:37
Holiday Schedule	Buses will operate on a holiday schedule on May 29th.	2023-05-05 18:00:00
Announcement 2	msg	2023-05-05 20:49:01

Shuttle Bus Schedule

Route ID	Route Name	Origin	Destination	Stops	Estimated Time	Bus Number
1	Red Route	Main Campus	Downtown Campus	Stop 1, Stop 2, Stop 3	00:30:00	1
2	Blue Route	North Campus	South Campus	Stop 1, Stop 2, Stop 3, Stop 4	01:00:00	2
3	Green Route	East Campus	West Campus	Stop 1, Stop 2, Stop 3, Stop 4	01:15:00	3

Privacy Policy Terms of Service © 2023 University Shuttle Bus App. All Rights Reserved.

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

The screenshot shows a web browser window with a dark theme. The address bar indicates the URL is `localhost:8000/driver_dashboard.php`. The page title is "Driver Dashboard". The header includes links for "Driver Dashboard", "Home", "About", "FAQs", a prominent blue "MAKE ANNOUNCEMENT" button, and "Log out". Below the header is a table displaying three shuttle routes:

Route ID	Route Name	Origin	Destination	Stops	Estimated Arrival Time	Bus Number
1	Red Route	Main Campus	Downtown Campus	Stop 1, Stop 2, Stop 3	00:30:00	1
2	Blue Route	North Campus	South Campus	Stop 1, Stop 2, Stop 3, Stop 4	01:00:00	2
3	Green Route	East Campus	West Campus	Stop 1, Stop 2, Stop 3, Stop 4	01:15:00	3

At the bottom of the page, there is a footer with small text: "Privacy Policy Terms of Service © 2023 University Shuttle Bus App. All Rights Reserved."

In the user/driver modules, we see the routes available to the user and driver. Our goal is to implement features so that the user can subscribe to different routes as needed in this dashboard. Drivers can send an announcement. Drivers can also update their current position via announcements. Like users, Drivers can subscribe to routes. Users can receive announcements from drivers managing

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

routes. User's won't receive route-specific announcements as they come in as a pop-up, but will see them in the announcements table. Other users can view the current positions of the shuttles. Users can also view the waiting time on any stoppage. We set this functionality up to have an active clock and an Estimated Time of arrival column to show exactly how far away the bus will arrive at a particular stop. Please see Mishell Quispe's section regarding driver and user functionality regarding announcements.

Description of Interface Requirements

- Allocation Type: Web-based/Desktop/Mobile
- Front-end software: HTML/CSS/JavaScript/Bootstrap/PHP
- Back-end software: Apache, MySQL, PHP, Bash
- Operating System: Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1024-aws x86_64)
- Supported Browsers: Mozilla/Chrome/Safari/Edge/Opera
- Editors and Compiler used: Vim, Nano, Visual Studio Code

Description of Nonfunctional Requirements

The screenshot shows a dark-themed login form titled "Log in". It has two input fields: "Email" and "Password". The "Email" field contains the value "fasdf". A validation message is displayed above the "Password" field: "Please include an '@' in the email address. 'fasdf' is missing an '@'." Below the form are two buttons: "Login" (blue) and "Register" (green).

Figure 7 Input Validation

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

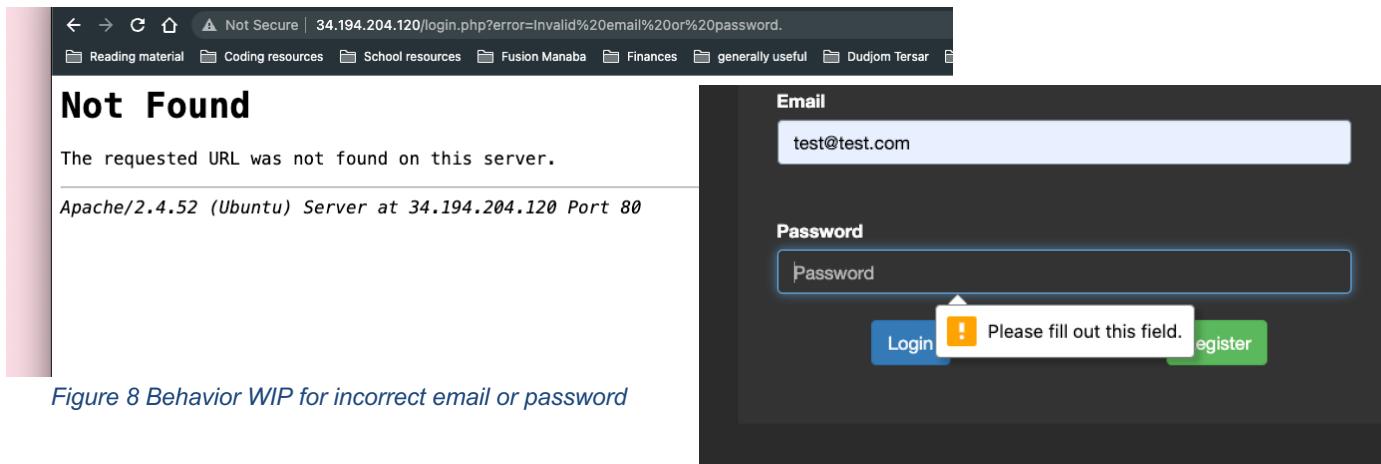


Figure 8 Behavior WIP for incorrect email or password

The above interactions with the website in production shows tests for Field Entry Validation on emails and passwords, before checking for the hash. Below is code that demonstrates the use of the “required” field in

HTML:

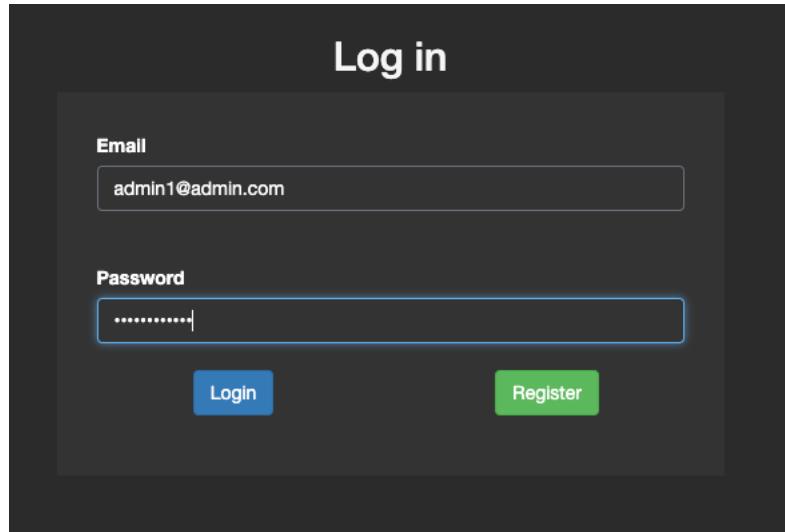
```
67     </div>
68     <!-- Login section -->
69     <h2 class="text-center">Log in</h2>
70     <div class="form-container">
71         <form class="form-group" method="POST" action="/authenticate.php">
72             <label for="email">Email &nbsp;&nbsp;</label>
73             <input type="email" class="form-control" id="email" name="email" placeholder="Enter
email" required>
74             <br>
75             <br>
76             <label for="password">Password</label>
77             <input type="password" class="form-control" id="password" name="password"
placeholder="Password" required>
78             <br>
79             <div class="text-center">
80                 <div class="row">
81                     <div class="col-md-6">
82                         <button type="submit" class="btn btn-primary d-inline-block mb-2 mr-md-2">Login</
button>
83                     </div>
84                     <div class="col-md-6">
85                         <a href="./registration.php" class="btn btn-success d-inline-block mb-2">Register</
a>
86                     </div>
87                 </div>
88             </div>
89         </form>
90     </div>
91     </div>
92 </main>
```

Once the user clicks the “Login” button, it will use this as an output to verify credentials against the MySQL database using a number of PHP scripts. The biggest contender that will constantly test active, registered users for their secure Enum “user_type”. If logged in user falls into one of three buckets, they will land in a specific dashboard tailored to the user. Please see the authenticate.php code that performs these operations when you click the login button.

Richie Molina, Mishell Quispe,
Danny Salas, Renee Rodas,
Victor Lluguicota

Dr. Kazi Zakia Sultana
CSIT 415-03 SP23
05/09/23

Software Engineering II: Project Report
Group 6: University Shuttle Bus App



Below you will see the redirect functionality testing the output PHP gets and compares it against the columns in the users table.

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

```
18
19
20 // Store email and password from the submitted form
21 $email = $_POST['email'];
22 $password = $_POST['password'];
23
24 // Sanitize email and password
25 $email = mysqli_real_escape_string($db_connection, $email);
26 $password = mysqli_real_escape_string($db_connection, $password);
27
28 // Prepare and execute the query to get the user information
29 $query = "SELECT * FROM users WHERE email = '$email' LIMIT 1";
30 $result = mysqli_query($db_connection, $query);
31
32 // Check if a user with the given email exists
33 if (mysqli_num_rows($result) == 1) {
34     $user = mysqli_fetch_assoc($result);
35
36     // Verify the submitted password with the stored hashed password
37     if (password_verify($password, $user['password'])) {
38
39         // Store user information in the session
40         $_SESSION['user_id'] = $user['id'];
41         $_SESSION['username'] = $user['username'];
42         $_SESSION['email'] = $user['email'];
43         $_SESSION['user_type'] = $user['user_type'];
44
45
46
47         // Redirect based on user type
48         if ($user['user_type'] == "user") {
49             header("Location: user_dashboard.php");
50             exit();
51         } else if ($user['user_type'] == "driver") {
52             header("Location: driver_dashboard.php");
53             exit();
54         } else {
55             header("Location: admin_dashboard.php");
56             exit();
57         }
58     } else {
59         // Password is incorrect
60         header("Location: /login.php?error=Invalid email or password.");
61         exit();
62     }
63 } else {
64     // Email doesn't exist
65     header("Location: /login.php?error=Invalid email or password.");
66     exit();
67 }
68
69 // Close the database connection
70 mysqli_close($db_connection);
71 ?>
72
```

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

To log into the development environment, we have enabled a security key that is a required parameter when using ssh to connect. Please see the steps below to securely access the development environment

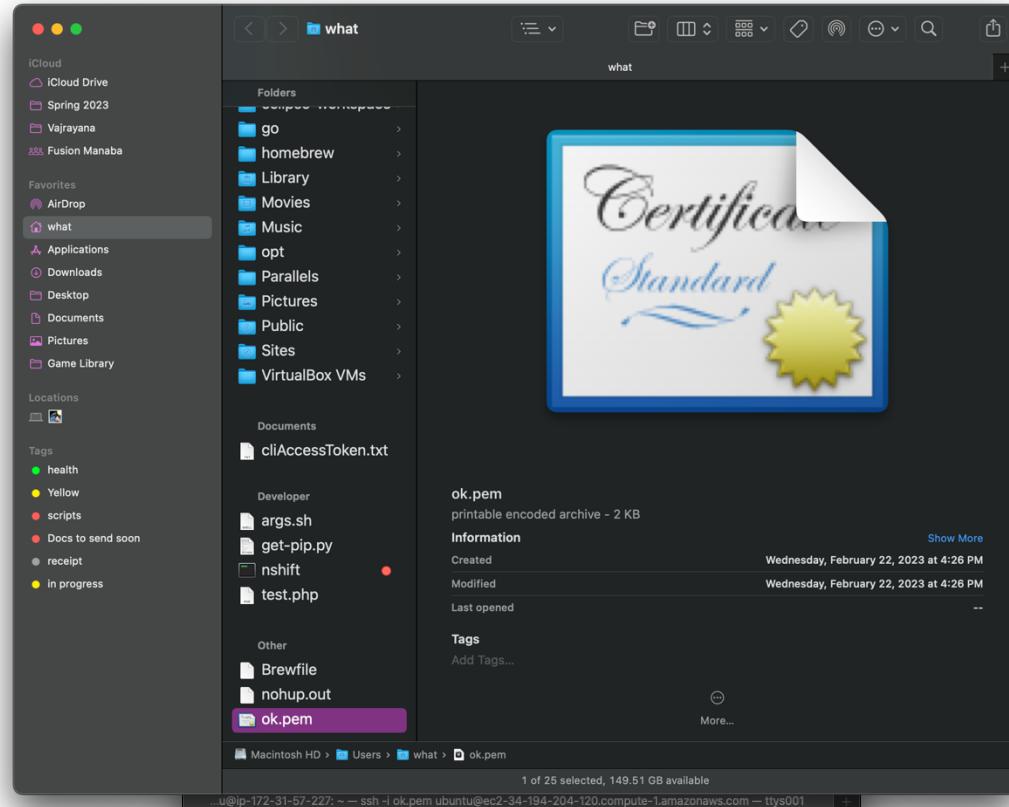


Figure 9 AWS EC2 Instance security Key used in ssh below

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Wed May 10 21:29:54 UTC 2023

System load: 0.013671875   Processes:          131
Usage of /: 17.9% of 28.89GB  Users logged in:      0
Memory usage: 27%           IPv4 address for eth0: 172.31.57.227
Swap usage:  0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

5 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Fri May  5 19:22:40 2023 from 130.68.183.14
ubuntu@ip-172-31-57-227:~$
```

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

Roles of the Team Members

- Project Manager: Richie Molina
- Front-End Developers: Richie Molina, Mishell Quispe, Danny Salas, Victor Lliguicota, Renee Rodas
- Back-end Developers: Richie Molina, Mishell Quispe, Danny Salas
- Quality Assurance Analysts: Danny Salas, Renee Rodas, Richie Molina, Victor Lliguicota
- Documentation Team: Danny Salas, Mishell Quispe, Richie Molina

Roles of the Team Members:

Mishell Quispe:

Admin's add/edit/delete routes module

Assumptions:

- only the admin can create and modify routes
- If the admin deletes a route from their end, it should also delete from the routes database
- There is no limit as to how many routes the admin can add
- Admin is permitted to enter an estimated arrival time for shuttles
- For every route, the admin has clearance to assign it to a bus number

Description:

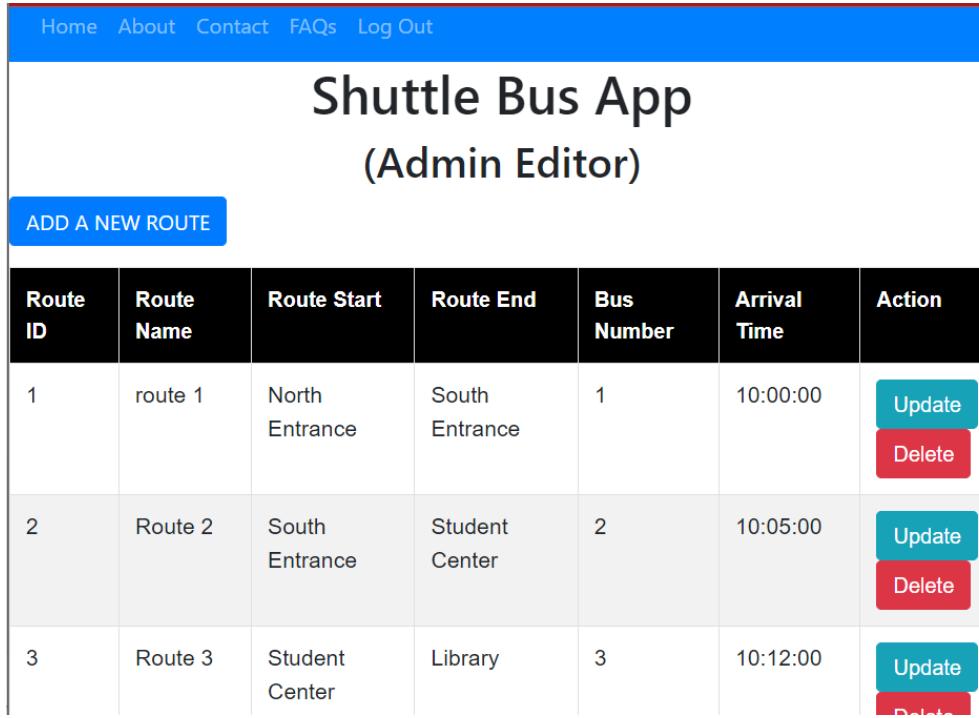
This process required the inclusion of five (5) files: index.php, insertproc.php, editproc.php, deleteproc.php and retrieve.php. The ‘retrieve.php’ is what displays the front end of the admin landing page with the table containing the routes’ id, name, starting point, destination, bus number and arrival time. In addition, the final column of the table has action options to update or delete each route. The display of the database table ‘routes’ is done by including the database file in the format of <?php include_once ‘db.php’; \$result = mysqli_query(\$conn, “SELECT * FROM routes”);?>. It firstly includes the already created database PHP file that connects the php to the database and then the mysqli_query() enables us to extract database information.

Afterward, we start the html ‘<html>’ to create a table ‘<table>’ and start with a table row ‘<tr>’ and the table headings ‘<th>’ with the aforementioned ‘routes’ columns. For the table rows to follow, we use the function \$row = mysqli_fetch_array(\$result) to obtain an array of the data in the ‘routes’ table from the specified row name in the following php lines:

```
<td><?php echo $row["id"]; ?></td>
<td><?php echo $row["route_name"]; ?></td>
<td><?php echo $row["route_start"]; ?></td>
<td><?php echo $row["route_end"]; ?></td>
<td><?php echo $row["bus_number"]; ?></td>
<td><?php echo $row["arrival_time"]; ?></td>
```

Software Engineering II: Project Report
 Group 6: University Shuttle Bus App

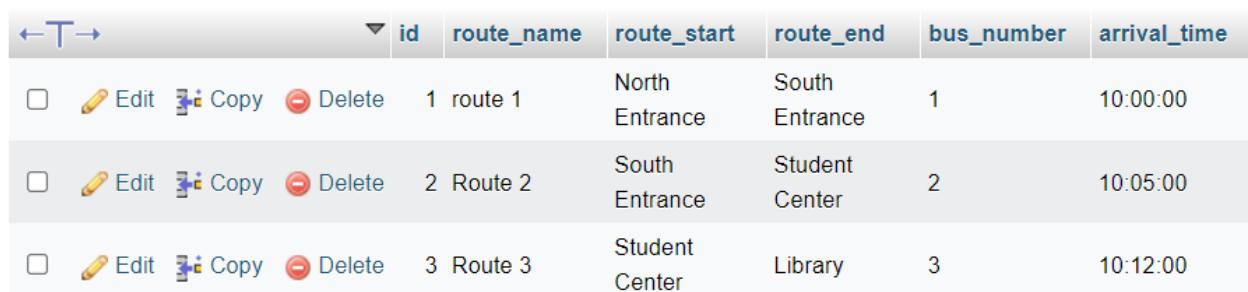
Admin Dashboard Screenshot



Route ID	Route Name	Route Start	Route End	Bus Number	Arrival Time	Action
1	route 1	North Entrance	South Entrance	1	10:00:00	<button>Update</button> <button>Delete</button>
2	Route 2	South Entrance	Student Center	2	10:05:00	<button>Update</button> <button>Delete</button>
3	Route 3	Student Center	Library	3	10:12:00	<button>Update</button> <button>Delete</button>

Figure 1: Admin dashboard from retrieve.php

Table ‘routes’



<input type="checkbox"/>	Edit	Copy	Delete	1	route 1	North Entrance	South Entrance	1
<input type="checkbox"/>	Edit	Copy	Delete	2	Route 2	South Entrance	Student Center	2
<input type="checkbox"/>	Edit	Copy	Delete	3	Route 3	Student Center	Library	3

Figure 2: ‘routes’ table in ‘shuttleroutes’ database

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

When the user selects “add a new route” from the homepage, they are redirected to the index form index.php. The ‘index.php’ and ‘insertproc.php’ work together, with the index being the HTML form to insert a new route and insertproc.php using the sql query to insert the inputs into the ‘routes’ table that follows the format \$sql = "INSERT INTO routes (id,route_name,route_start,route_end,bus_number,arrival_time) VALUES ('\$id','\$route_name','\$route_start','\$route_end','\$bus_number','\$arrival_time')";. The following screenshots show a new route entry and also a screenshot of what the database looks like

Insert Form



Shuttle Bus App

Insert a New Route

[Go Back to List](#)

Route ID:	<input type="text" value="4"/>
Route name:	<input type="text" value="Route 4"/>
Route start:	<input type="text" value="Library"/>
Route end:	<input type="text" value="Student Housing"/>
Bus number:	<input type="text" value="4"/>
Arrival time:	<input type="text" value="05:15 PM"/> <input type="button" value=""/>
<input type="button" value="submit"/>	

Figure 3: insert form in index.php with new route inputs

New Route in ‘routes’ Table

	<input type="button" value="←"/>	<input type="button" value="→"/>		<input type="button" value="▼"/>	id	route_name	route_start	route_end	bus_number	arrival_time
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	route 1	route 1	North Entrance	South Entrance	1	10:00:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	Route 2	Route 2	South Entrance	Student Center	2	10:05:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	Route 3	Route 3	Student Center	Library	3	10:12:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	Route 4	Route 4	Library	Student Housing	4	17:15:00

Figure 4: ‘routes’ table after admin enters a new route

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

New Route in Admin Dashboard

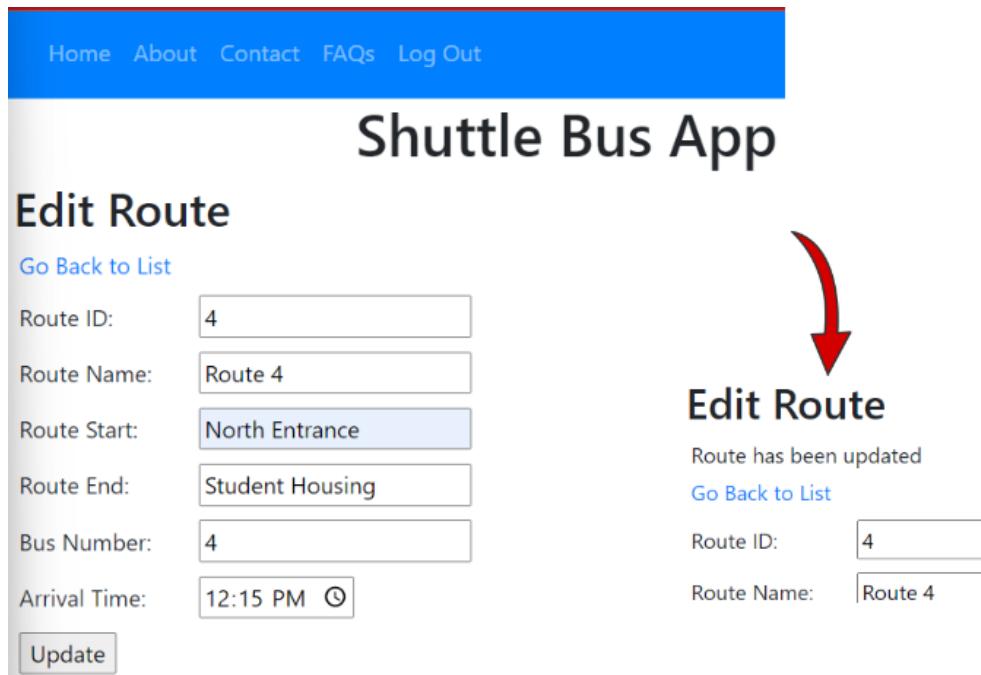
Shuttle Bus App (Admin Editor)						
ADD A NEW ROUTE						
Route ID	Route Name	Route Start	Route End	Bus Number	Arrival Time	Action
1	route 1	North Entrance	South Entrance	1	10:00:00	<button>Update</button> <button>Delete</button>
2	Route 2	South Entrance	Student Center	2	10:05:00	<button>Update</button> <button>Delete</button>
3	Route 3	Student Center	Library	3	10:12:00	<button>Update</button> <button>Delete</button>
4	Route 4	Library	Student Housing	4	17:15:00	<button>Update</button> <button>Delete</button>

Figure 5: retrieve page from admin side after route 4 has been entered

The ‘editproc.php’ can be accessed when the admin selects the “update” option under the action column. It displays the same form as the insert form and allows the admin to edit the details of any route using the format `mysqli_query($conn,"UPDATE routes set id='".$POST['id']."' , route_name='".$POST['route_name']."' , route_start='".$POST['route_start']."' , route_end='".$POST['route_end']."' , bus_number='".$POST['bus_number']."' , arrival_time='".$POST['arrival_time']."' WHERE id='".$POST['id']."');`. After the admin saves the changes, there is a confirmation message that says the route has been updated.

Edit Route 4

Software Engineering II: Project Report
 Group 6: University Shuttle Bus App



The screenshot shows a web application interface. At the top, there is a blue header bar with links: Home, About, Contact, FAQs, and Log Out. Below the header, the title "Shuttle Bus App" is displayed in a large, bold, dark font. Underneath the title, the heading "Edit Route" is shown in a bold, dark font. To the left of the main content area, there is a "Go Back to List" link. The main form contains fields for editing a route: "Route ID" (4), "Route Name" (Route 4), "Route Start" (North Entrance), "Route End" (Student Housing), "Bus Number" (4), and "Arrival Time" (12:15 PM). There is also a "Update" button. To the right of the main form, a red arrow points down to a second "Edit Route" section. This section displays a message "Route has been updated" and shows the updated route information: "Route ID" (4) and "Route Name" (Route 4).

Figure 6: edit form in ‘editproc.php’ that changes route start and the arrival time
 Updated Route 4

Route ID	Route Name	Route Start	Route End	Bus Number	Arrival Time	Action
1	route 1	North Entrance	South Entrance	1	10:00:00	<button style="background-color: #009640; color: white; border: none; padding: 5px;">Update</button> <button style="background-color: #dc3545; color: white; border: none; padding: 5px;">Delete</button>
2	Route 2	South Entrance	Student Center	2	10:05:00	<button style="background-color: #009640; color: white; border: none; padding: 5px;">Update</button> <button style="background-color: #dc3545; color: white; border: none; padding: 5px;">Delete</button>
3	Route 3	Student Center	Library	3	10:12:00	<button style="background-color: #009640; color: white; border: none; padding: 5px;">Update</button> <button style="background-color: #dc3545; color: white; border: none; padding: 5px;">Delete</button>
4	Route 4	North Entrance	Student Housing	4	12:15:00	<button style="background-color: #009640; color: white; border: none; padding: 5px;">Update</button> <button style="background-color: #dc3545; color: white; border: none; padding: 5px;">Delete</button>

Figure 7: Admin dashboard after admin modifies the route start and the arrival time
 The ‘deleteproc.php’ file uses the sql query \$sql = "DELETE FROM routes WHERE id="" .
 \$_GET["id"] . """; to delete a route based on the id of the route the admin selects for deletion.

Driver announcements module

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

Assumptions:

- only the driver can make announcements regarding accidents on route or simple delays
- Announcements are different than current position updates

Description:

The announcements creation process required the addition of five (5) files: announcements.php, announce_index.php, announce_addproc.php, announce_delete.php and announce_edit.php. This module follows the same structure as the admin's add/edit/delete functions. A summary of how it works is that the driver's announcement will be stored on the database table 'announcements', which is also used on the user's side to extract this information to display for the user. Whenever the driver wants to make an announcement, they must select the "make announcement" option on the navigation bar, which will redirect them to announcements.php. Here they will be able to see an announcements table with the history of all their announcements so unless they choose to delete them, past announcements will remain on the table as well. If the driver selects the "add an announcement" option, they will need to enter the announcement's title, announcement and date to delete the announcement. When the driver submits their inputs, they will be added in the 'announcements' table that has the columns 'id', 'title', 'message', 'addedOn' and 'deleteOn'. 'Id' is an auto-increment value and 'addedOn' automatically shows the current time in which the driver makes a message for the user.

Driver "Make Announcement" Form

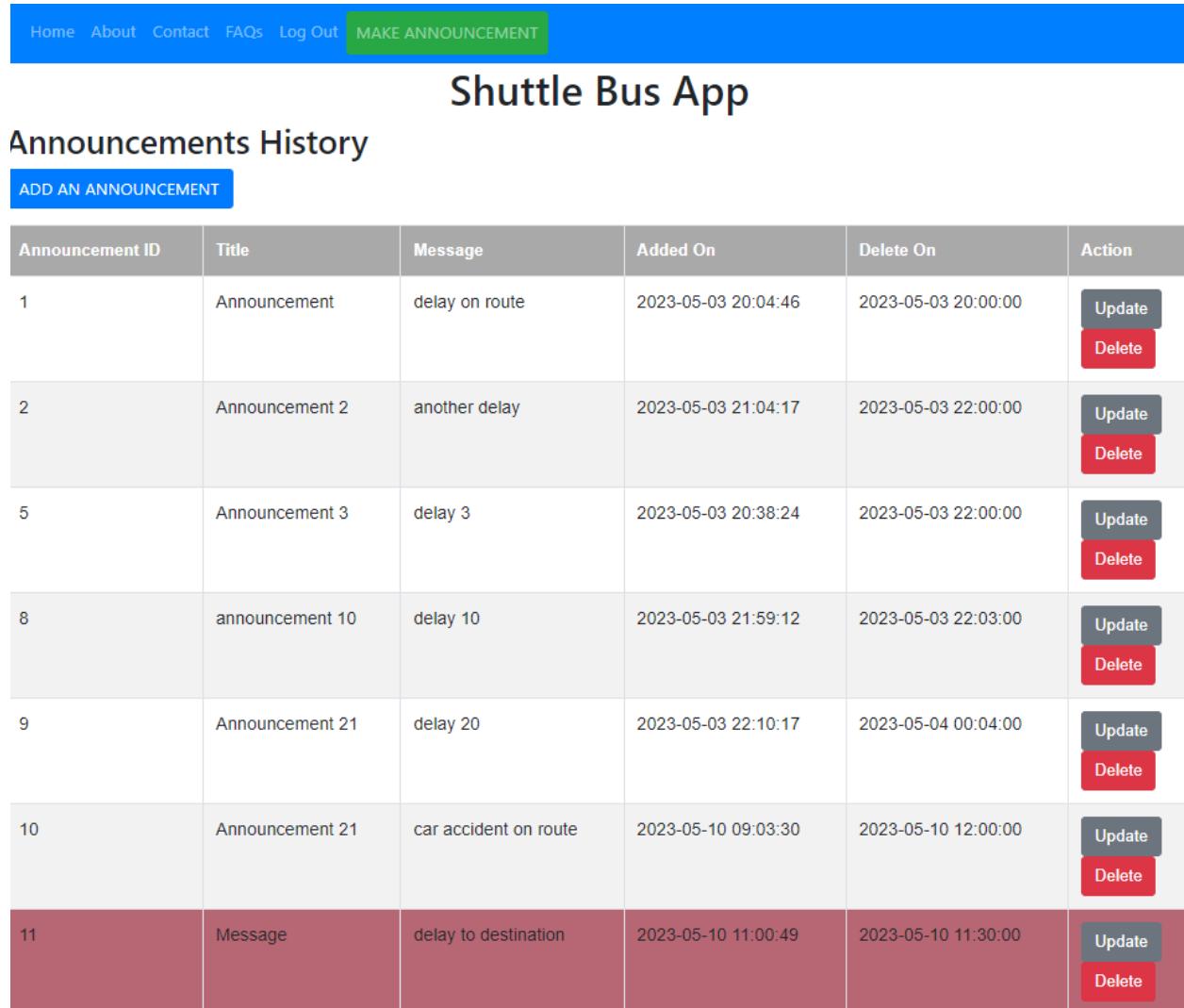
The screenshot shows a web-based application interface for a shuttle bus app. At the top, there is a blue header bar with white text containing links: Home, About, Contact, FAQs, Log Out, and a green button labeled "MAKE ANNOUNCEMENT". Below the header, the main title "Shuttle Bus App" is centered in a large, bold, dark font. Underneath the title, the section "Make an Announcement" is displayed in a larger, bold, dark font. To the left of the form fields, there is a link "See All Announcements". The form itself consists of several input fields and controls:

- A "Title:" label followed by a text input field containing the placeholder "Message".
- An "Announcement:" label followed by a text input field containing the placeholder "delay to destination".
- A "Added On:" label followed by the text "10-05-2023 @09:21:34am".
- A "Delete On:" label followed by a date and time input field showing "05/10/2023 11:30 AM" and a small calendar icon.
- A large "Enter" button at the bottom of the form.

Figure 8: New announcement "Message" created using form in driver dashboard

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

Driver Dashboard After New Announcement



The screenshot shows a driver dashboard with a blue header bar containing links for Home, About, Contact, FAQs, Log Out, and a green 'MAKE ANNOUNCEMENT' button. Below the header is a large title 'Shuttle Bus App'. Underneath the title is a section titled 'Announcements History' with a blue 'ADD AN ANNOUNCEMENT' button. The main content is a table with the following data:

Announcement ID	Title	Message	Added On	Delete On	Action
1	Announcement	delay on route	2023-05-03 20:04:46	2023-05-03 20:00:00	<button>Update</button> <button>Delete</button>
2	Announcement 2	another delay	2023-05-03 21:04:17	2023-05-03 22:00:00	<button>Update</button> <button>Delete</button>
5	Announcement 3	delay 3	2023-05-03 20:38:24	2023-05-03 22:00:00	<button>Update</button> <button>Delete</button>
8	announcement 10	delay 10	2023-05-03 21:59:12	2023-05-03 22:03:00	<button>Update</button> <button>Delete</button>
9	Announcement 21	delay 20	2023-05-03 22:10:17	2023-05-04 00:04:00	<button>Update</button> <button>Delete</button>
10	Announcement 21	car accident on route	2023-05-10 09:03:30	2023-05-10 12:00:00	<button>Update</button> <button>Delete</button>
11	Message	delay to destination	2023-05-10 11:00:49	2023-05-10 11:30:00	<button>Update</button> <button>Delete</button>

Figure 9: New announcement “Message” on driver’s announcement history

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

User Dashboard with Announcements

The screenshot shows the top portion of the user dashboard. At the top, there is a blue header bar with links for Home, About, Contact, FAQs, and Log Out. Below the header, the title "Shuttle Bus App" is centered. Underneath the title, the heading "User Dashboard" is displayed. A yellow announcement box contains two entries:

- Announcement 21**: car accident on route. Updated made on 2023-05-10 09:03:30.
- Message**: delay to destination. Updated made on 2023-05-10 11:00:49.

Figure 10: Top Portion of user dashboard with the announcements

The announce_edit.php file allows the driver to edit their announcement. In the form instead of there being a space for the date in which an announcement was added on, there is an “updated on” input that automatically changes to the current time that the driver is updating the announcement. For example, in the image below, the driver edits the announcement and time to delete the message on. When the driver selects “update”, the message and delete date are updated, as well as the time it was updated. The user side is also updated with the new message and the time and date that it happened.

“Edit Announcement” Form

The screenshot shows the "Edit Announcement" form. At the top, there is a blue header bar with links for Home, About, Contact, FAQs, Log Out, and a green button labeled "MAKE ANNOUNCEMENT". Below the header, the title "Shuttle Bus App" is centered. The main section is titled "Edit Announcement". It includes a link "See All Announcements" and a table for editing an announcement:

Announcements	
ID:	11
Title:	Message
Announcement:	delay to destination by 15
Updated On:	05/10/2023 11:00:49 AM <input type="button" value=""/>
Delete On:	05/10/2023 11:20 AM <input type="button" value=""/>

At the bottom of the form is a blue "Update" button.

Figure 11: Driver edits announcement 11 for the announcement and delete time and date

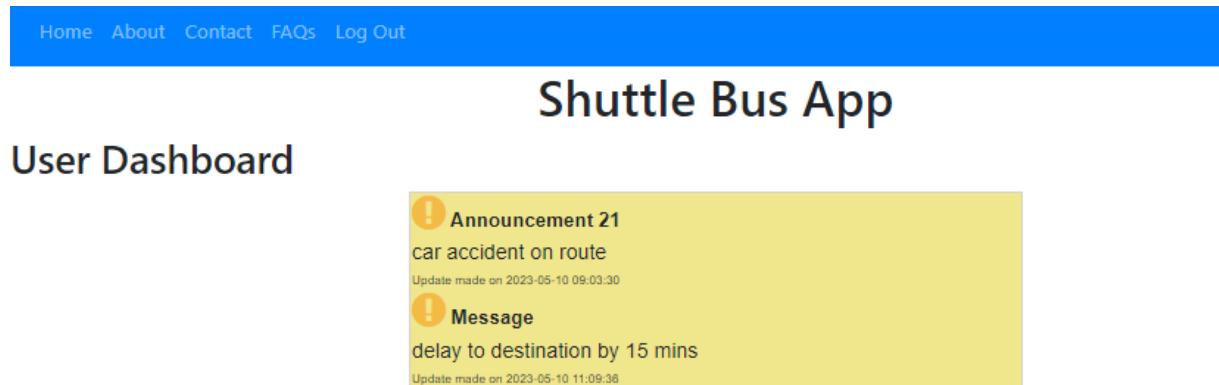
Software Engineering II: Project Report
Group 6: University Shuttle Bus App

Announcements Table After Edit

Announcement ID	Title	Message	Added On	Delete On	Action
11	Message	delay to destination by 15 mins	2023-05-10 11:09:36	2023-05-10 11:20:00	<button>Update</button> <button>Delete</button>

Figure 12: Announcement table after announcement 11 was edited

User Dashboard After Edit



The screenshot shows a user dashboard for the "Shuttle Bus App". At the top, there is a blue header bar with links for Home, About, Contact, FAQs, and Log Out. Below the header, the title "Shuttle Bus App" is centered. Underneath the title, the heading "User Dashboard" is displayed. A prominent yellow box contains two announcements. The first announcement, titled "Announcement 21", states "car accident on route" and includes a timestamp "Update made on 2023-05-10 09:03:30". The second announcement, titled "Message", states "delay to destination by 15 mins" and includes a timestamp "Update made on 2023-05-10 11:09:36".

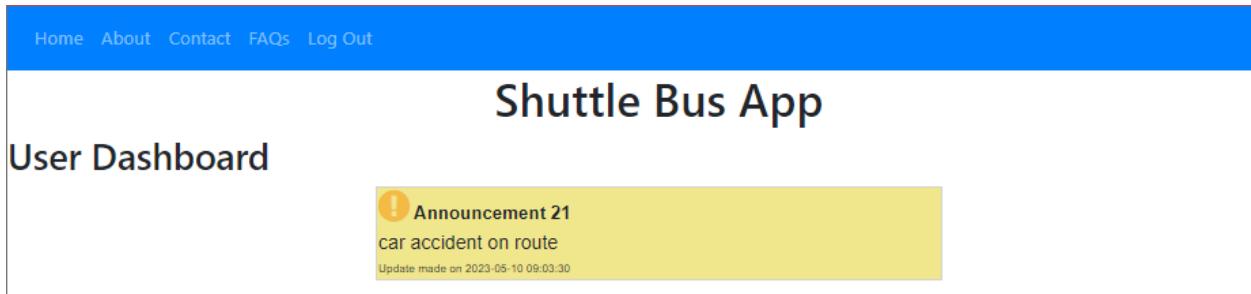
Figure 13: User dashboard after announcement 11 was edited

The announce_delete.php file works the same way as the deleteproc.php file for the routes' function. Only the driver can delete the announcement from the 'announcements' table.

The “deleteOn” column is an important part of the web application as it's used to only display the announcements that the driver wants the users to see until the time/date it's set to be deleted. On top of the user_dashboard.php file, we added the following query: \$result = mysqli_query(\$conn,"SELECT * FROM announcements WHERE deleteOn >= NOW()");. This means that for the user side, only those announcements that have a “deleteOn” beyond the current time are retrieved. The “deleteOn” date/time being less than the current date/time means that it will no longer show on the user dashboard. In the image below, the current time is 11:30:00, which means that announcement 11 will not show to the user. However, the announcement will remain on the announcements history table on the driver dashboard, as well as in the ‘announcements’ database table.

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

User Dashboard Without “Message”



The screenshot shows a user dashboard for the "Shuttle Bus App". At the top, there is a blue header bar with links for Home, About, Contact, FAQs, and Log Out. Below the header, the title "Shuttle Bus App" is centered above the "User Dashboard" section. In the "User Dashboard" section, there is a yellow rectangular box containing an exclamation mark icon, the text "Announcement 21", and the message "car accident on route". Below the message, it says "Update made on 2023-05-10 09:03:30".

Figure 14: User dashboard after announcement 11 was set to “delete” at 11:20:00 with the current time being 11:30:00

Database Table ‘announcements’

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	title	message	addedOn	deleteOn
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	Announcement	delay on route	2023-05-03 20:04:46	2023-05-03 20:00:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	Announcement 2	another delay	2023-05-03 21:04:17	2023-05-03 22:00:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	Announcement 3	delay 3	2023-05-03 20:38:24	2023-05-03 22:00:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	announcement 10	delay 10	2023-05-03 21:59:12	2023-05-03 22:03:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9	Announcement 21	delay 20	2023-05-03 22:10:17	2023-05-04 00:04:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	Announcement 21	car accident on route	2023-05-10 09:03:30	2023-05-10 12:00:00
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	Message	delay to destination by 15 mins	2023-05-10 11:09:36	2023-05-10 11:20:00

Figure 15: Database table ‘announcements’ still with all the announcements

Roles of the Team Members:

Danny Salas

My duty for this project was figuring out how to add an announcement table, and have an announcement system. Our goal was to have an input for the driver to inset announcements, and it would be able to be seen by the students. It was important for the students to view the announcements because it was a demonstration that there is communication between student and driver. We used PHP and HTML in order for the announcements to work. First we had multiple php files that were made to have the add function and the delete function. Both the add and delete php files were made to have their own functions and work properly. Our goal was to have an announcement bar as clear as it can be, in order for the website to be the most user friendly it can be. The most difficult part of this part I encountered was being able to implement everything together and having it show in the website. The queries were also a tough task I encountered because I was never familiar with it. Being able to store everything in the database was something new and I enjoyed working with databases because it did teach me a lot. For example there was a part of the announcement php,

```
<td><?php echo $row["announce_id"]; ?></td>
<td><?php echo $row["title"]; ?></td>
<td><?php echo $row["message"]; ?></td>
<td><?php echo $row["addedOn"]; ?></td>
<td><?php echo $row["deleteOn"]; ?></td>
<td><a href="announce_edit.php?id=<?php echo $row["announce_id"]; ?>" class="btn btn-secondary">Update</a>
<br>
<a href="announce_delete.php?announce_id=<?php echo $row["announce_id"]; ?>" class="btn btn-danger" onclick="return confirm('Are you sure you want to delete this announcement?')>Delete</a>
```

This was created in order to display the table of announcements and here we made the rows that belonged to each task. So we had the title column, the message column where the message would be displayed. And The added and deleted column where it would show the date when this happened. Little by little we were able to create the code that would help us put it on the website. Our announcements were able to work perfectly, and it led to us assembling the whole website in one.

My experience working with partners was great. I really enjoyed working with people I never met before. I also find it very helpful this way. I do not think I would of been able to finish this

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

project by myself. Coding can be stressful so being able to have partners to help is great in order to finish it, especially under being under a time constraint. Our group had great communication, and we were able to ask each other what we needed help with. We were also able to meet on campus and work together which helped a lot. Although our project did not result in the most perfect way, it was something to learn. It is never the best idea to edit things right before the deadline and we committed this mistake. Time management is crucial and I feel like we may of needed to of handled it better, which would of resulted in better results.

Non - Function Requirements :

Some non - functional requirements we used were security, password hashing, maintainability, and usability. Non-functional requirements are essential to a software project's success since it can significantly improve the system's overall quality. For example, a website with no security are more opened to receive cyber attacks, or data leaks. This is a problem we see in the news from time to time because people end up losing their own information on websites they thought were secure. A website with low security may also frequently shut down from time to time, which ends up affecting reviews and might have a negative effect on user experience and productivity.

For our project, we were able to have password hashing. Password hashing is being able to protect the user's passwords from being read, if there is any type of data leak. We believed that having hashed password was crucial and would receive positive feedback from users. Password hashing is crucial, making it safe for the user, while also protecting information and limiting the amount of cyber attacks and leaking information.

Roles of the Team Members:

Renee Rodas

I was involved in contributing to making the bus tables and the HTML code that contained the stops, locations, time, and schedules. I also worked on creating a few number of query processes to improve the user experience. Also Implemented a feature that allowed the bus timetable to be reversed and applied it to every bus but was not able to be implemented to the project due to the time. Additionally, I played a small role in developing the database with buses and contributed to the driver announcement system HTML code and PHP that sends updates to users.

Project Experience

While working on the project simultaneously proved to be a huge challenge for me because I had no prior knowledge with PHP or HTML, it was also a lot of fun for me to learn about PHP queries. However, I thought the experience was rewarding and fun.

Software Engineering II: Project Report
Group 6: University Shuttle Bus App

Roles of the Team Members:

Victor Lliguicota

Description of the Functional Requirements:

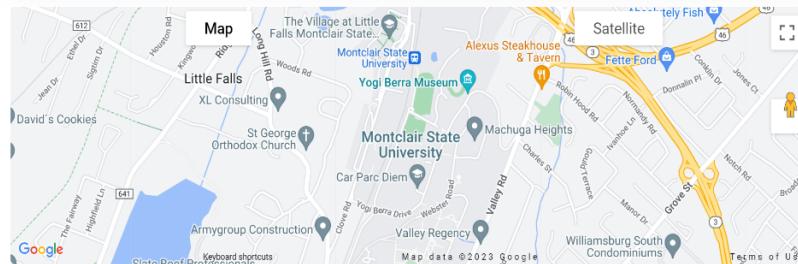
Admin Module:

View Current Positions of the Shuttles

5/10/2023, 3:46:39 PM

Montclair State University Shuttle Management Schedule

Arrival	Estimated Arrival	Shuttle Route	Current Shuttle Stop	Next Shuttle Stop	Shuttle Status	Shuttle ID	Driver ID
9:15 AM	30 Minutes	Route A	University Hall	CarPare Diem	Operational	0010	1
10:15 AM	25 Minutes	Route R/H	Red Hawk Garage	University Hall	Operational	0011	2
9:30 AM	10 Minutes	Route T	Sinatra Hall	Hawk Crossings	Operational	0012	3
11:15 AM	5 Minutes	Route V	Fenwick Hall	Basic Hall	Operational	0013	4
12:30 AM	15 Minutes	Route V/T	Lot 60	NJ Transit Station	Operational	0014	5



Software Engineering II: Project Report
 Group 6: University Shuttle Bus App

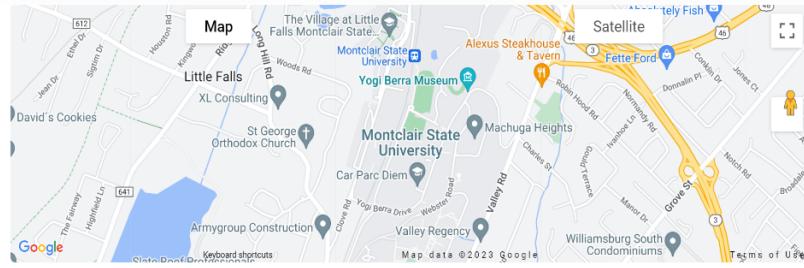
User Module:

Other users can view current position of the shuttle

5/10/2023, 3:46:39 PM

Montclair State University Shuttle Management Schedule

Arrival	Estimated Arrival	Shuttle Route	Current Shuttle Stop	Next Shuttle Stop	Shuttle Status	Shuttle ID	Driver ID
9:15 AM	30 Minutes	Route A	University Hall	CarParc Diem	Operational	0010	1
10:15 AM	25 Minutes	Route R/H	Red Hawk Garage	University Hall	Operational	0011	2
9:30 AM	10 Minutes	Route T	Sinatra Hall	Hawk Crossings	Operational	0012	3
11:15 AM	5 Minutes	Route V	Fenwick Hall	Basic Hall	Operational	0013	4
12:30 AM	15 Minutes	Route V/T	Lot 60	NJ Transit Station	Operational	0014	5

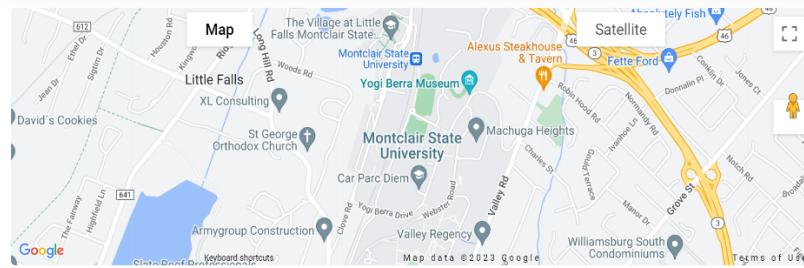


Users can also view the waiting time in any stoppage

5/10/2023, 3:46:39 PM

Montclair State University Shuttle Management Schedule

Arrival	Estimated Arrival	Shuttle Route	Current Shuttle Stop	Next Shuttle Stop	Shuttle Status	Shuttle ID	Driver ID
9:15 AM	30 Minutes	Route A	University Hall	CarParc Diem	Operational	0010	1
10:15 AM	25 Minutes	Route R/H	Red Hawk Garage	University Hall	Operational	0011	2
9:30 AM	10 Minutes	Route T	Sinatra Hall	Hawk Crossings	Operational	0012	3
11:15 AM	5 Minutes	Route V	Fenwick Hall	Basic Hall	Operational	0013	4
12:30 AM	15 Minutes	Route V/T	Lot 60	NJ Transit Station	Operational	0014	5



Non-Functional Requirements:

Security:

For security we decided to implement form validation and password hashing. For form validation we decided that the Admin and User should have different permissions. The Admin is the only user that can Add/Delete/Edit the routes when needed for the benefit of the system. And the User can only access and interact with the route schedules and route alerts. The way we implemented form validation was to assign different login type information to Admin and User. This way the login type must match Admin to login as an Admin user, and the login type must match User to login as a User. For password hashing we used the SHA1 algorithm that MySQL provides for both the Admin and User passwords. This hashing makes the passwords unknown in case the server is compromised and keeps account information safe.

Maintainability:

For maintainability we decided to use AWS (Amazon Web Services) to keep the website running and available to users at any time. The advantage of using AWS is that we can have reliable access to web servers all the time and can fix any errors that users are experiencing in no time at all because the web servers are up and running whenever the developers need to fix or provide new updates/modifications to the source code.

Roles of the Team Members:

Group 6 Team Members:

Victor Lliguicota was assigned:

- Selection of Non-Functional Requirements:

The Non-Functional Requirements are Security and Maintainability we decided to use these requirements because we wanted the user to have confidence when using our system and come back to the system because it is always working and reliable. For security, we implemented we decided that the users should have their accounts and should only be able to access their accounts with assigned credentials. Each user was given a username and password so that only they can view and access their information and profile. We decided to implement different account permissions so that an Admin user is the only one that can manipulate the system and modify it if needed. And the User can only view and interact with the system without modifying the system. For passwords, a hashing algorithm was implemented to protect user information from malware and server vulnerabilities.

- Implementation of Shuttle Schedule/Tracking/Estimated Time of Arrival

The Shuttle schedule shows the current shuttles that are active and operational to the User and keeps updating to let the User know when the shuttle arrives. The shuttle is tracked with the Google Maps API, which helps track the shuttle's position and helps determine how far away the shuttle is from your current position. Also, the API will calculate any ETA from any stop and show the time to the User. The API is connected to the database and can be modified according to the Admin if a new stop is needed or a route needs modified.

Roles of the Team Members:

Richie Molina

I was in charge of getting the foundation of everything up and running. Since I had the most experience with a LAMP stack, we decided to try out the security features AWS offered to the free-mid tier Ubuntu instances. I also managed the Github organization and repository along with implementing the prototype via static Github Pages. We merged over to a LAMP stack once we established an Elastic IP address since whenever we rebooted the server, we would receive a new IP address. From then, we could build off a static IP address that delivered dynamic requests. I also built the landing page, user dashboards, about page, FAQs, Privacy Policy, and Terms of Service pages, established the large-text enabled dark mode theme via a global stylesheet for production, provided documentation and credentials for development access, coordinated check in meetings, implemented code from group members into the PHP stack by ensuring correct file redirections occur, appropriating linkages, securing table headers align with group member's PHP/MySQL query, and ultimately condensing dashboards into a Singleton design pattern for each dashboard user type.

Project Experience

The experience gained from working on this University Shuttle Bus App was a great learning experience for our team. We were able to come together despite our differences in skillset to create a functioning dynamic website living on the internet just like every other website out there! If we can do this much, we can create anything. Learning to schedule time for certain portions of code with the help of design patterns helped streamline the process by knowing why to start with the database first. Although towards the deadline we ran into issues with PHP scripts and MySQL table headers close to demo presentation time, we were able to clearly explain functionality and code at a conversational level. I have only interacted with the output of MySQL queries in the form of reports and Excel workbooks. While I compared the query against the table headers, I began to understand how important it is to settle on table headers way before you start to implement PHP scripts that for example compare the existing users table's specific column headers such as first name or email and password. If we are attempting to join tables to create a view for the student user or admin user, it is important to denote headers with purposeful naming conventions. As we learned to work through these hurdles as a group through the feature implementation phase from the work provided by Mishell Quispe and Victor Lliguicota for example, aligning headers and file name conventions before and during development through clear communication is key.