

Live Track

BUS MANAGEMENT AND LIVE TRACKING SYSTEM

Live Bus Tracking – IOT Project

Overview

This IoT-based bus tracking web and application is designed to enhance the convenience and efficiency of university transport for students. The system integrates GPS modules with MongoDB for real-time location tracking, an admin panel for centralized management, and a user-friendly interface for commuters.

Key Feature

The primary feature of this app is **real-time bus tracking**, allowing students to monitor bus movements and plan their commutes effectively. This reduces waiting times and enhances overall transport efficiency.

System Architecture

1. **IoT Integration:** *GPS module* installed on buses sends real-time location data to a MongoDB.
2. **Database:** *MongoDB* is used for storing bus locations, user data, routes, stops, and administrative records.
3. **Web Interface:** Frontend is built using React.js for dynamic user experience. And Backend is on Node.js/Express.js is used to handle API requests and communicate with MongoDB.

Features

Admin Panel

Admins can add, edit, and remove:

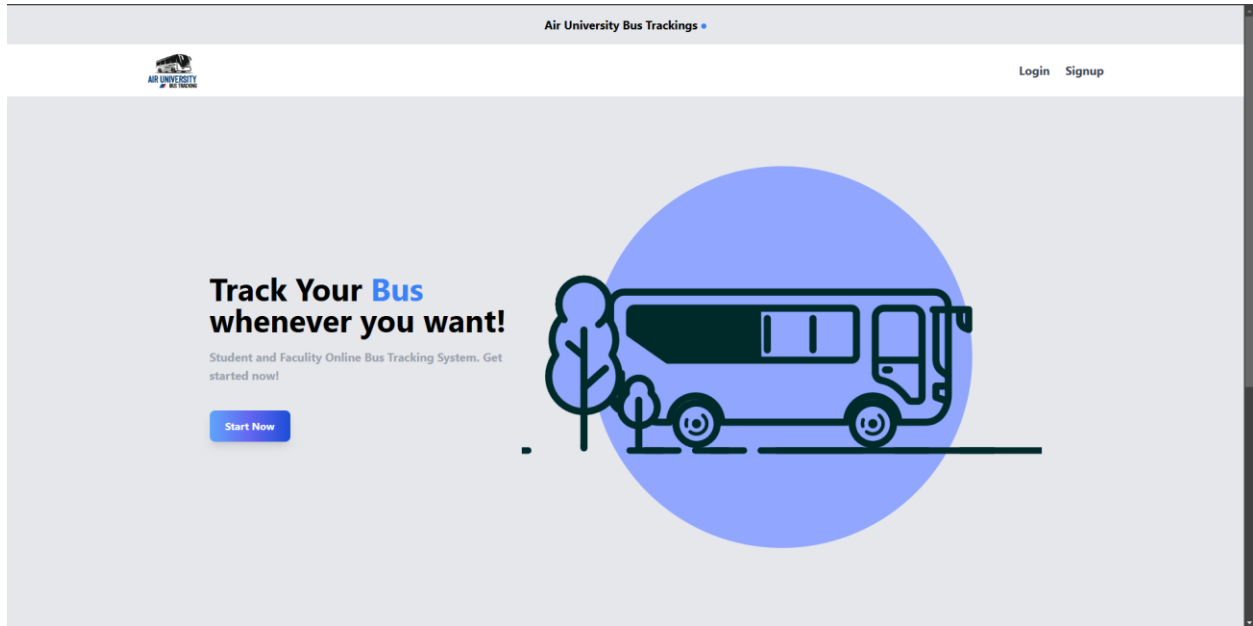
- **Buses:** Maintain buses records.
- **Drivers:** Maintain driver records linked to buses.
- **Routes & Stops:** Manage routes and stops assigned to buses.
- **Users:** Control students access and details.

User End

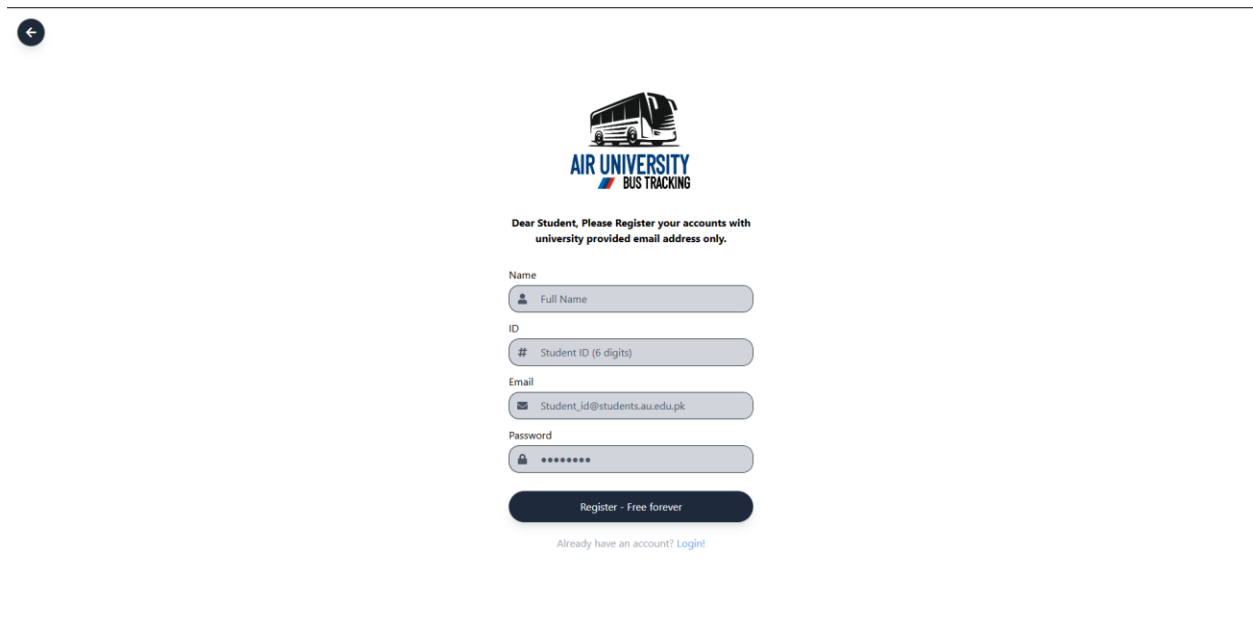
- **User Registration & Login:** Secure authentication for personalized experiences.
- **Profile Management:** Users can add personal details and preferred stops for tracking.
- **Live Location Tracking:** Real-time display of bus location on **Google Maps**.
Integration with user-specific stop data for an enhanced tracking experience.

Website Interface – “Abdullah Tareen – 210690”


Landing Page.




Signup Page / Registration Page



Login Page.





Dear Student, Please enter your Student Email and Password to login.



Email

Password

Login


[Forgot Password? Reset Here!](#)
[Don't have an account? Sign Up!](#)

Home Page / User Dashboard where user can see their selected stops, and can track live location of the bus.


HOME STOPS 

Air University Bus Tracking

Route: Wah
Stop: NewCity
[Click here to see your Bus schedule](#)

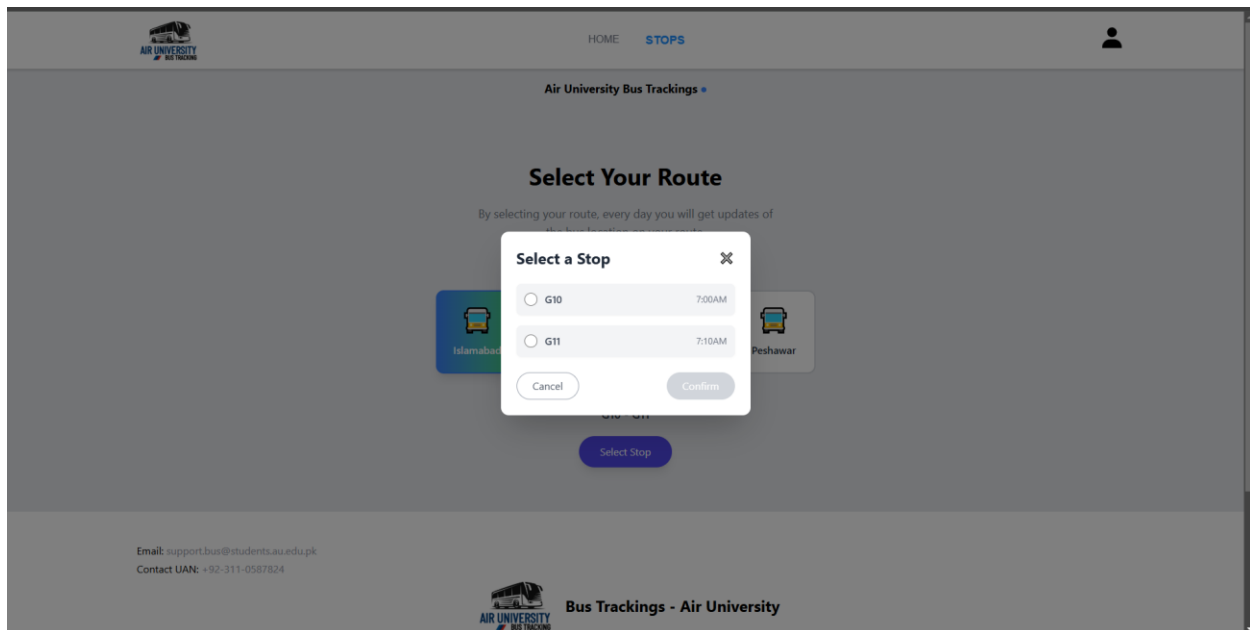
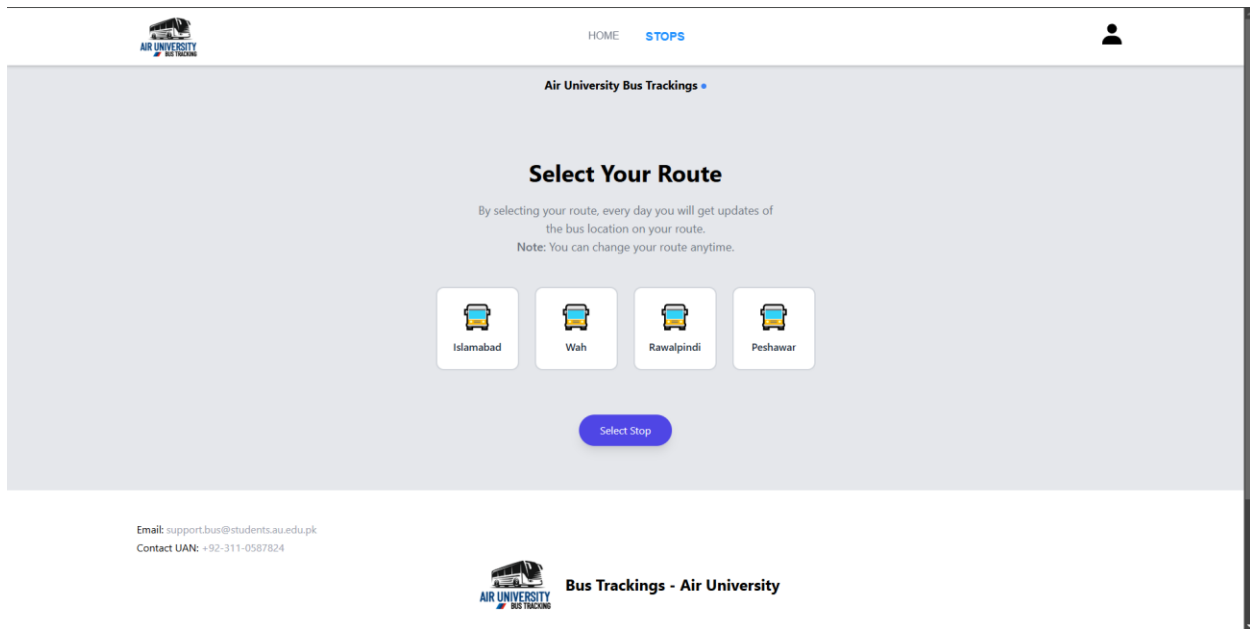


Email: support.bus@students.au.edu.pk
Contact UAN: +92-311-0587824


**Bus Trackings - Air University**


Install app from

Allowing user to change their stop.




Allowing user to add or edit their personal details.






Abdullah
210690@students.au.edu.pk


Cell No

 03110587824

Parent Phone

 03345499493

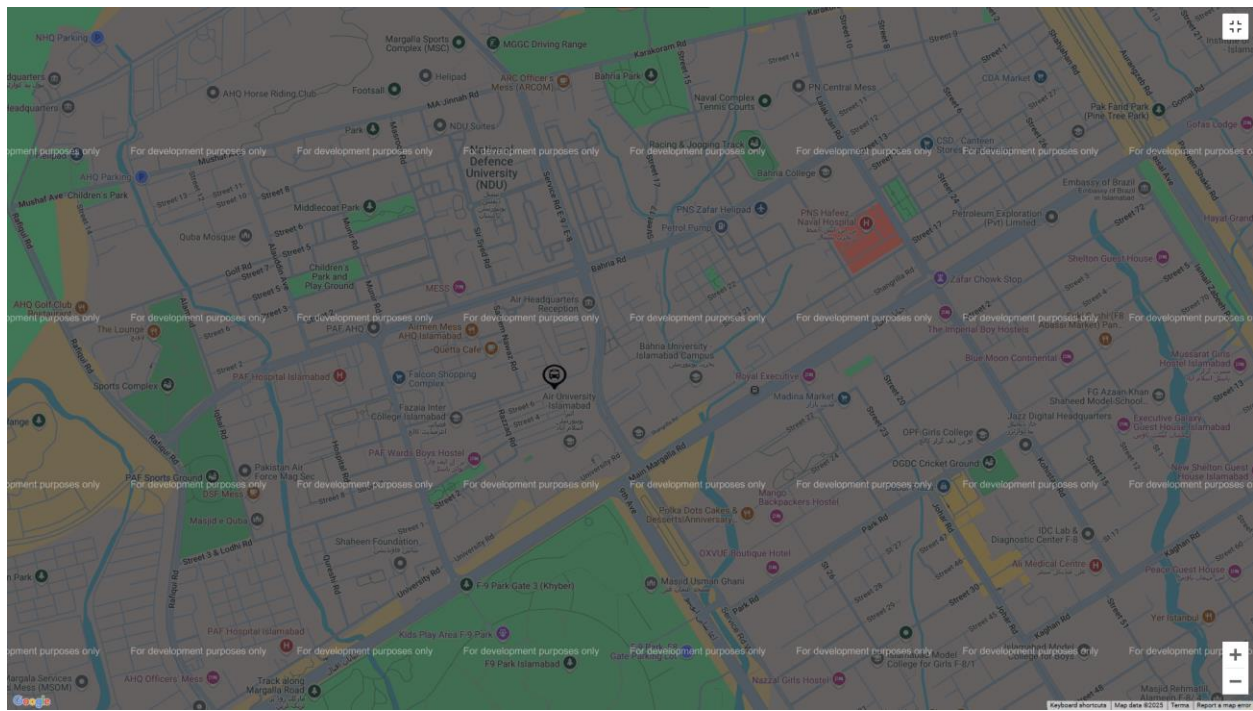
Address

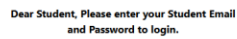
 D-293 1/2

Update

Logout

Allowing user to track Live location of their Bus.





Email

 adminabdullah@gmail.com

Password

.....

[Login](#)

Forgot Password? [Reset Here!](#)

Don't have an account? [Sign Up!](#)

Admin can Add users and see existing user details, with the right to delete a user.

Bus Management

MAIN

- Dashboard

MANAGEMENT

- Students
- Buses
- Drivers
- Bus Routes
- Assignment

SETTINGS

- Settings
- Logout

Students

Add New Student

Add Student

| Name | Email | Phone | Parent Phone | Address | Route | Stop | Actions |
|----------------|---------------------------|-------------|--------------|-----------|-------|---------|-------------------------|
| Abdullah | 210690@students.au.edu.pk | 03110587824 | 03345499493 | D-293 1/2 | Wah | NewCity | <button>Delete</button> |
| Ayesha Shahzad | 210680@students.au.edu.pk | N/A | N/A | N/A | Wah | Basti | <button>Delete</button> |

Bus Management

Add New Bus

Admin can Add Buses and see existing bus details, with the right to delete or modify any bus details.

Bus Management

Ayesha Shahzad 210680@students.au.edu.pk N/A N/A N/A Wah Basti Delete

MAIN

- Dashboard

MANAGEMENT

- Students
- Buses
- Drivers
- Bus Routes
- Assignment

SETTINGS

- Settings
- Logout

Bus Management

Add New Bus

Add Bus

| Bus Number | Registration | Chassis | Actions |
|------------|--------------|--------------|-----------------------------------------------|
| 3A | 89 | 1289812891 | <button>Edit</button> <button>Delete</button> |
| 4F | tf-09 | 54454545 | <button>Edit</button> <button>Delete</button> |
| 1B | LPT-8675 | 312456781263 | <button>Edit</button> <button>Delete</button> |

Drivers

Admin can Add Drivers and see existing driver details, with the right to delete or modify any driver details.

Bus Management

MAIN

Dashboard

MANAGEMENT

Students

Buses

Drivers

Bus Routes

Assignment

SETTINGS

Settings

Logout

Drivers

Add New Driver

Name

ID Card Number

Years of Experience

Age

Phone Number

Add Driver

| Name | ID Card | Experience | Age | Phone | Actions |
|-------------|-----------|------------|-----|-------------|-----------------------------------|
| Saim | 123456789 | 2 | 24 | 03115331153 | <div>Edit</div> <div>Delete</div> |
| Abdullah | 12345678 | 5 | 32 | 1234567 | <div>Edit</div> <div>Delete</div> |
| Sanaullah x | 123456789 | 12 | 60 | 23456754 | <div>Edit</div> <div>Delete</div> |

Admin can Add Routes and Stops against each route, see existing route details, with the right to delete or modify any the details.

Admin can Assign Buses with Drivers and Routes, each driver and route can only be assigned to one bus at a time. Admin can see existing assignment details, with the right to assign any bus with any of the available drivers and routes.

The screenshot shows the 'Assigned Routes and Duties' section of the Bus Management system. The interface includes a sidebar with navigation options: MAIN (Dashboard), MANAGEMENT (Students, Buses, Drivers, Bus Routes, Assignment), and SETTINGS (Settings, Logout). The main content area displays a table with the following data:

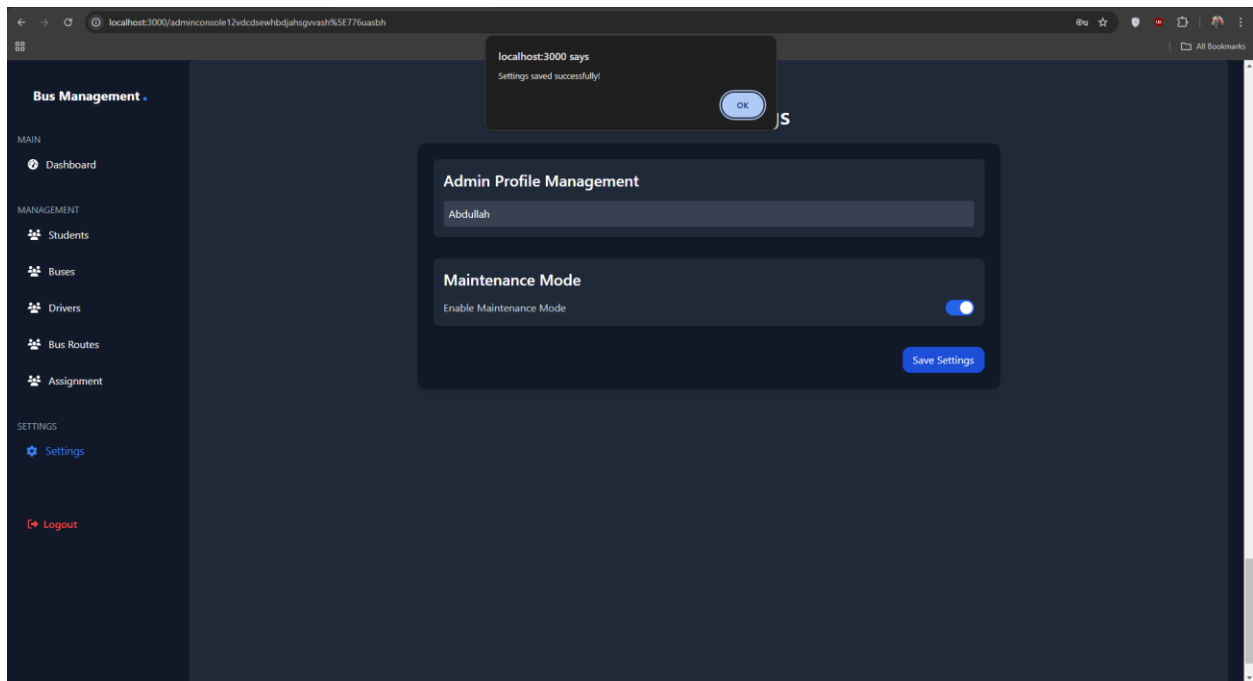
| Bus Number | Driver Assigned | Route Assigned | Actions |
|------------|-----------------|----------------|----------------------|
| 3A | Abdullah | Islamabad | Edit |
| 4F | Saim | Wah | Edit |
| 1B | Sanaullah x | Rawalpindi | Edit |

Below the table, there is an 'Admin Settings' section with two sub-sections: 'Admin Profile Management' showing the name 'Abdullah' and 'Maintenance Mode'.

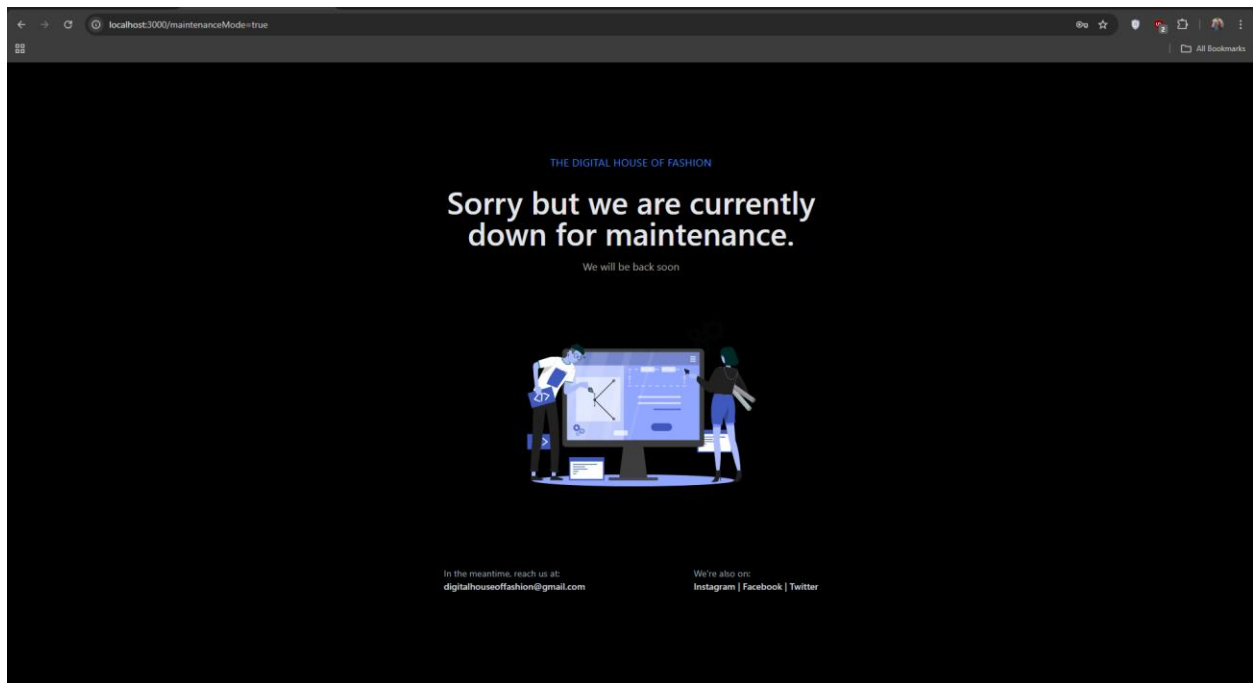
Admin can change Admin name which will be displayed on the top, and can also put the website into maintenance mode.

The screenshot shows the 'Admin Settings' section of the Bus Management system. The interface includes a sidebar with navigation options: MAIN (Dashboard), MANAGEMENT (Students, Buses, Drivers, Bus Routes, Assignment), and SETTINGS (Settings, Logout). The main content area displays the 'Admin Settings' section with two sub-sections: 'Admin Profile Management' showing the name 'Abdullah' and 'Maintenance Mode' with a toggle switch labeled 'Enable Maintenance Mode'. A 'Save Settings' button is located at the bottom right of the settings area.

Maintenance mode demo... Let's turn on the Maintenance mode...

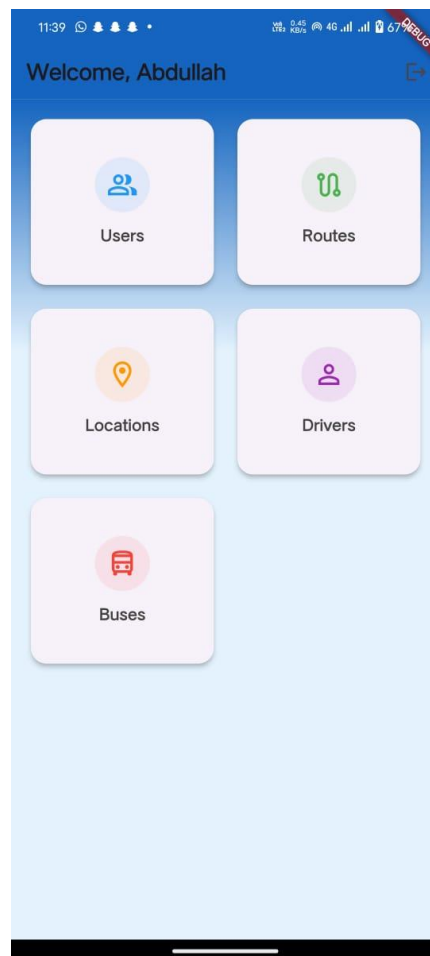
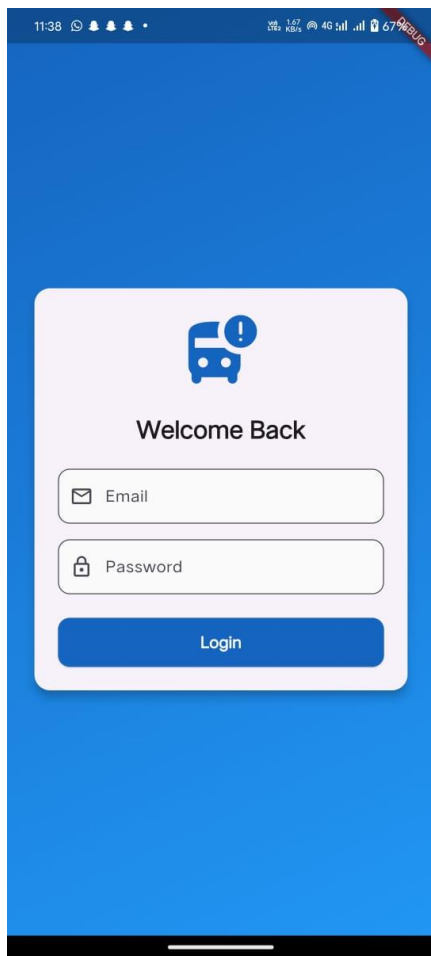


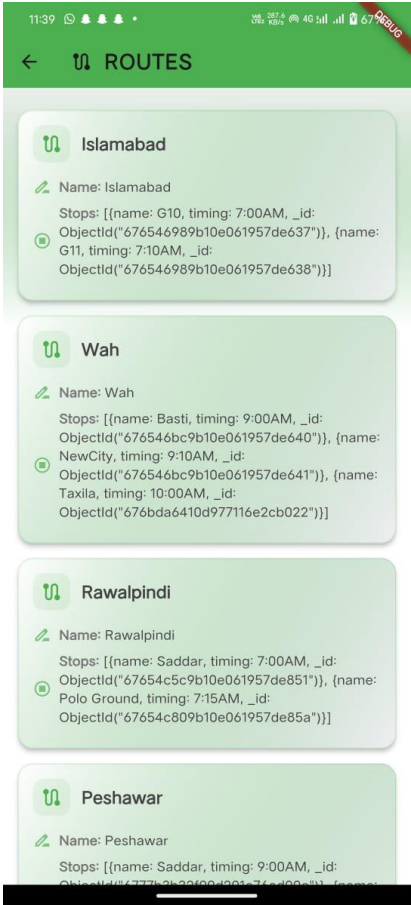
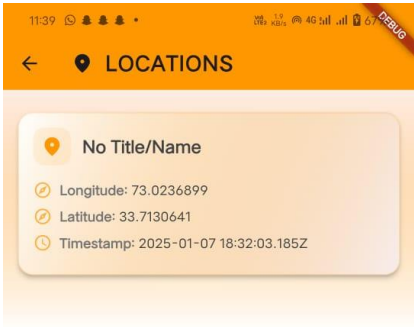
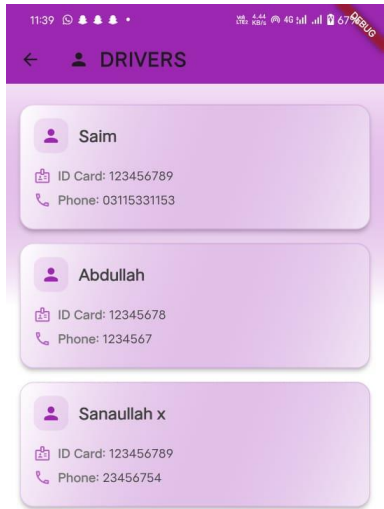
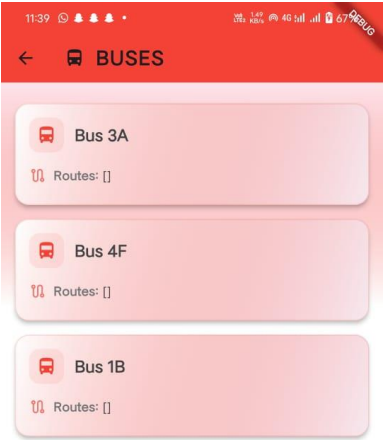
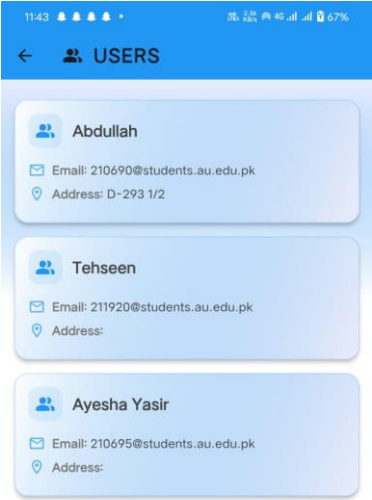
Now let's access the website... It should not be accessible anymore until admin turns off the maintenance mode.



Application Interface

Now let's access the mobile application... Student can login using the details existing inside Database. If not, user must have to register first using Web. And then student see the details.





Arduino and Hardware

Arduino sending location coordinates via GPS Tracker.

The screenshot displays the Arduino IDE interface. At the top, the menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for opening files, saving, compiling, uploading, and monitoring the serial port.

The main editor window shows the following C++ code:

```
#include <inyGPS.h>  
#include <SoftwareSerial.h>  
#include <ESP8266WiFi.h>  
#include <WiFiClientSecure.h>
```

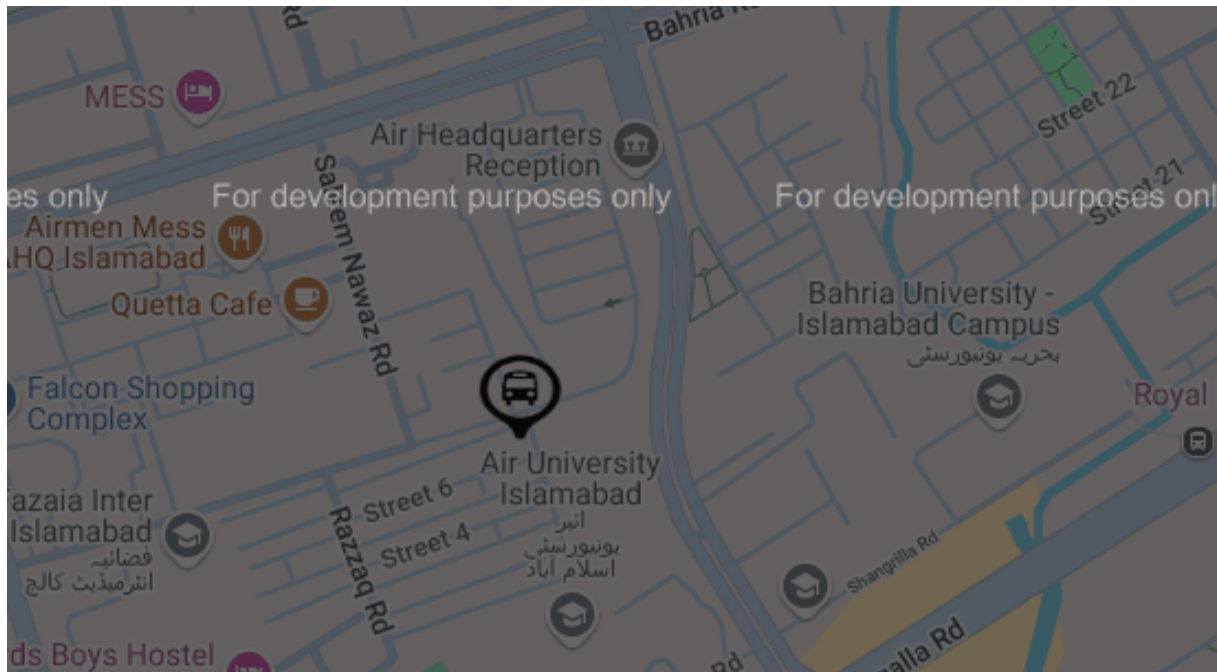
The Serial Monitor window at the bottom shows the output of the program. It begins with a message from the NodeMCU 0.9 (ESP-12 Module) on COM5, indicating it is waiting for a valid location. After several attempts, it successfully connects to a WiFi network named "WIFI connected!".

The IP address is 192.168.79.166, and the synchronization time with NTP is shown. The time is synchronized, and the program continues to wait for a valid location. Subsequently, it sends a payload to a server, receives a successful response, and updates its location. The location update is confirmed by the "Location updated successfully!" message, which includes the latitude and longitude coordinates.

Data received at Mongo DB.

```
_id: ObjectId('677d7323a10eb00c90f33152')
latitude: "33.7130641"
longitude: "73.0236899"
timestamp: 2025-01-07T18:32:03.185+00:00
__v: 0
```

Location fetching from Mongo and Displaying on the Website.



Hardware Prototype.

