RouteFlow: Iteration 1 Report

Project Team

Hamd ul Haq 23I-0081
Saif Shahzad 23I-2634
Haider Abbas 23I-2558

Department of Artificial Intelligence and Data Science

National University of Computer and Emerging Sciences
Islamabad, Pakistan

,

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Transportation is an essential part of student life, ensuring accessibility and convenience for daily commutes to university and home. However, many university bus providers (like in FAST) donot provide a centralized platform for students to manage their transportation needs. These challenges often lead to uncertainty, delays, and missed buses, causing inconvenience for students who rely on these services.

RouteFlow is designed to streamline the bus transportation system by introducing a dynamic, real-time platform that allows students to access updated bus schedules, manage their travel plans efficiently, and receive instant notifications regarding any changes. The system focuses on enhancing user experience by integrating features such as secure authentication, interactive dashboards, and automated schedule updates, ensuring a seamless and well-organized transportation service.

Sprint 1 marks the initial phase of **RouteFlow**'s development, laying the foundation for core functionalities. This sprint focuses on implementing a secure login and registration system, enabling students to access their accounts and transportation details efficiently.The sprint includes the development of a structured dashboard that serves as the primary interface for users, providing quick access to essential information. By the end of this sprint, the fundamental architecture of the application will be established, ensuring a solid framework for future enhancements in subsequent iterations.

Chapter 2

Sprint 1: User Stories

User stories serve as the foundation for understanding the key functionalities that need to be implemented in RouteFlow. In this sprint, we focused on essential features such as authentication, bus schedule access, and user interaction with the dashboard. Each user story is broken down into sub-user stories to ensure clarity in requirements and implementation.

2.1 User Authentication (Login/Signup)

User Story: As a student, I want to log in using my credentials so that I can access my bus transportation account.

Sub-User Stories:

- If incorrect credentials are entered, an error message should be displayed to help the user correct their mistake.
- A show/hide password option should be available while typing to ensure accuracy.
- The session should remain active until the user logs out to avoid repeated logins.

User Story: As a new student, I want to sign up by providing my university details so that I can create an account.

Sub-User Stories:

- The system should validate the university ID entered during signup for verification purposes.
- After signing up, the system should send a confirmation email to verify the account.
- The password should meet security requirements to ensure account safety.

2.2 Password Reset Functionality

User Story: As a student, I want to reset my password if I forget it so that I can regain access to my account.

Sub-User Stories:

- The system should send a password reset link to the registered email to allow secure password recovery.
- The new password should meet security standards to ensure protection.
- A confirmation message should be displayed after a successful password reset.

2.3 Dashboard and Navigation

User Story: As a student, I want to access the home page after logging in so that I can navigate through the bus reservation system.

Sub-User Stories:

- The home page should display a personalized welcome message with the user's name.
- Quick access to bus schedules and routes should be available on the dashboard.
- The system should provide a clear and intuitive layout for easy navigation.

User Story: As a student, I want a dashboard that provides an overview of my transportation details so that I can quickly access important information.

Sub-User Stories:

- The dashboard should display the registered bus route and timings.
- A sidebar menu should be available for smooth navigation between different sections.
- A dark mode option should be provided for a customizable user experience.

2.4 Secure Logout Functionality

User Story: As a student, I want a logout button on the dashboard so that I can securely log out of my transportation account.

Sub-User Stories:

- A logout confirmation prompt should appear to prevent accidental logouts.
- After logout, the system should redirect the user to the login page.
- The session should be terminated completely after logging out to ensure account security.

Chapter 3

Structured Specifications

This section outlines the structured specifications for the key functionalities implemented in Sprint 1. Each specification describes the functionality, input/output requirements, and relevant constraints.

3.1 Login Authentication

- **Functionality:** Secure user authentication via Roll number and password.
- **Input:** Roll number (format: i(batch no)(XXXX)), password.
- **Output:** Dashboard access if authentication is successful; an error message if credentials are incorrect.
- **Constraints:**
  - Session management should ensure automatic logout after a period of inactivity.
  - Passwords should be stored securely using salted hashing.
  - Multi-factor authentication (MFA) should be integrated for enhanced security.

3.2 User Registration

- **Functionality:** Allows new users to create an account using their university credentials.
- **Input:** Full name, Roll number, password

- **Output:** A confirmation email.
- **Constraints:**
  - Password must meet security requirements (at 4 to 8 characters).
  - Duplicate email addresses should not be allowed.

3.3 Password Reset

- **Functionality:** Enables users to reset their password if they forget it.
- **Input:** Email request for a reset link.
- **Output:** A password reset form provided that the email is correct and valid within the given database.
- **Constraints:**
  - Secure password hashing and validation.
  - Users must provide a new password that meets security requirements.

3.4 Bus Schedule Display

- **Functionality:** Displays real-time bus schedules based on the selected route.
- **Input:** User-selected route or bus number.
- **Output:** Display of current schedule, estimated arrival times, and delays.
- **Constraints:**
  - Schedule data should be updated dynamically.
  - Users should be able to filter by time, route, and destination.
  - The system should provide alerts for unexpected delays or cancellations.

3.5 Dashboard and Navigation

- **Functionality:** Provides users with an overview of transportation details and quick access to schedules.
- **Input:** User login credentials.

- **Output:** A dashboard displaying bus schedules, routes, and real-time notifications.
- **Constraints:**
  - The interface should be user-friendly and responsive.
  - Personalized welcome messages should be displayed based on user login data.

3.6 Secure Logout

- **Functionality:** Ensures secure session termination upon logout.
- **Input:** Logout request from the user.
- **Output:** Redirection to the login page with the session fully terminated.
- **Constraints:**
  - Active session data should be removed upon logout.
  - Users should receive a confirmation prompt before logging out.
  - The session should expire automatically after a period of inactivity.

Chapter 4

Sprint Workflow Management via Scrum Board

In this sprint, we utilized **Trello** to manage our Scrum workflow effectively. Trello helped us organize tasks, track progress, and ensure that all deliverables were completed within the sprint timeline. Below, we provide a detailed breakdown of our sprint activities along with relevant snapshots.

4.1 Snapshot 1: Sprint Planning

During the initial sprint planning phase, we identified the key features to be implemented and created a structured backlog. The following tasks were defined:

- Created and populated the sprint backlog with user stories.
- Defined sprint objectives, including authentication, dashboard navigation, and bus schedule integration.
- Assigned tasks to team members based on priority and complexity.
- Set up the workflow structure in Trello, categorizing tasks into *Sprint Backlog, In Progress, Done, Testing,* and *Leftover*.

4.2 Snapshot 2: Mid-Sprint Progress

At the mid-sprint stage, substantial progress had been made on core functionalities. The development workflow was actively progressing, and tasks were moved between columns based on their completion status. The following milestones were achieved:

- **Login Authentication:** Successfully implemented student authentication functionality.
- **Dashboard Navigation:** Initial UI and navigation system developed.
- **Password Reset Feature:** Implemented email-based password reset functionality.
- **Bus Schedule Display:** Integration of real-time bus schedules was underway.

4.3 Snapshot 3: Sprint Completion

By the end of the sprint, most of the planned user stories were implemented, and a testing phase was initiated to validate functionalities. The final sprint outcomes were:

- **Login Authentication:** Fully tested and functional for student access.
- **Dashboard and Navigation:** Completed with a smooth user interface.
- **Password Reset Feature:** Validated with secure email recovery process.
- **Signup Functionality:** Successfully integrated with the authentication system.
- **Secure Logout:** Deferred to the next sprint due to priority adjustments.

4.4 Conclusion

The sprint was successfully executed with significant progress on core functionalities. Although one feature (Secure Logout) was left for the next sprint, the main objectives of this sprint were achieved. Moving forward, feedback from testing will be incorporated, and the remaining feature will be prioritized in the upcoming sprint.

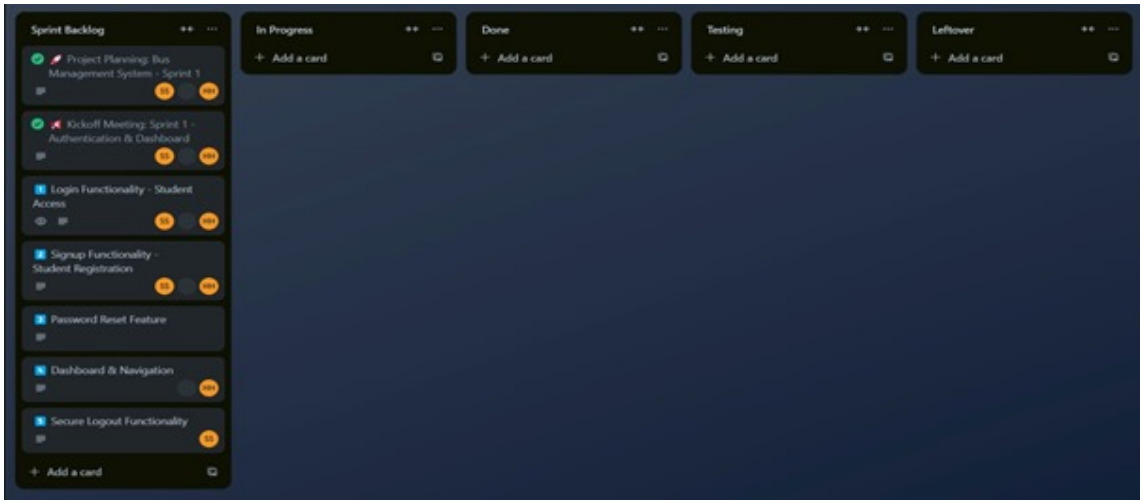

Figure 4.1: Sprint Planning Phase: Task Assignment and Backlog Setup

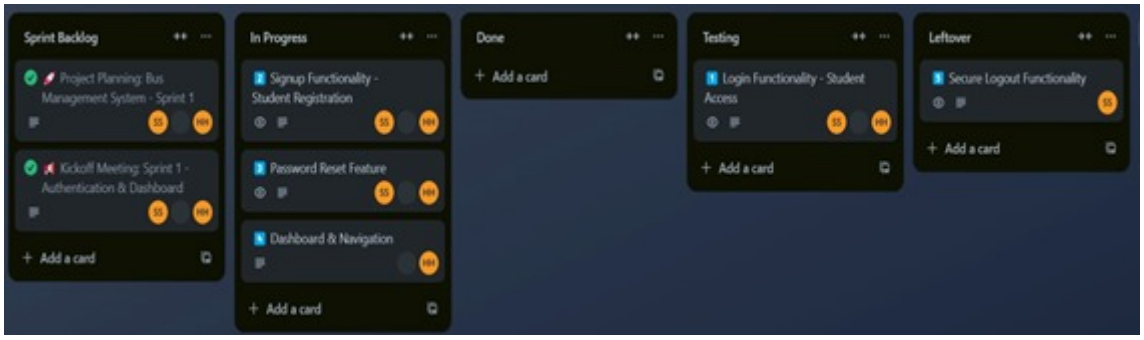

Figure 4.2: Mid-Sprint Progress: Active Development and Task Movement

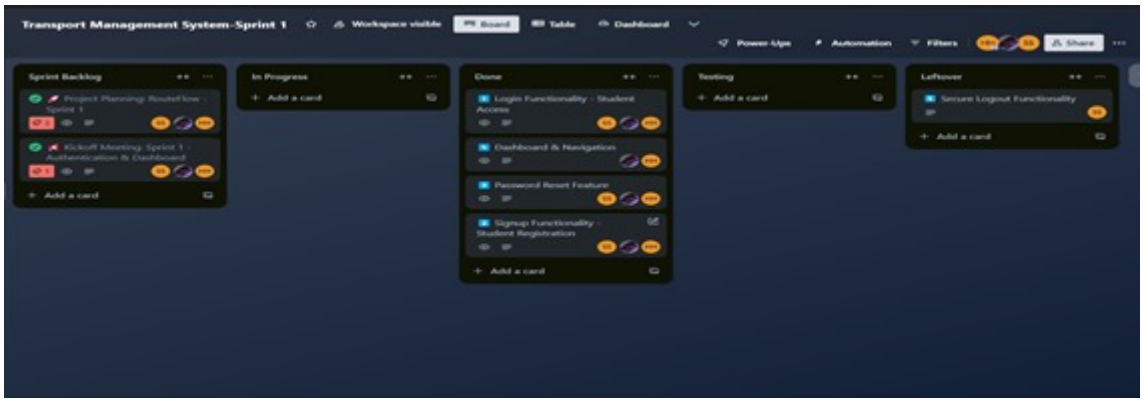

Figure 4.3: Sprint Completion: Task Completion and Testing Phase

Chapter 5

Non-Functional Requirements (NFRs) of RouteFlow

Non-functional requirements define system attributes such as performance, security, usability, and reliability. These requirements ensure that the system functions optimally under expected workloads and provide a seamless experience for users. Below is a breakdown of the non-functional requirements specifically for Sprint 1: Basic UI & User Authentication.We will strive to formally implement these non-functional requirements through rigorous design, testing, and validation to ensure the system meets performance, security, and usability standards as outlined in Sprint 1.

Category	Requirement Description
Performance	The login and signup functionalities should process user authentication requests within 2 seconds under normal load conditions. The dashboard UI should load within 3 seconds.
Availability	The authentication system should be available 99.9% of the time, ensuring users can log in or sign up without major downtime.
Security	User passwords should be **securely stored** using hashing and salting. Important user data should be **encrypted** for protection. The system should **block login attempts** for some time if a user enters the wrong password **5 times in a row** to prevent hacking attempts.
Usability	The system should be **easy to use** and work well on **both computers and mobile devices**. The design should be **clear and simple** so that students can easily log in and sign up. It should also follow basic **accessibility guidelines** to help all users navigate the system comfortably.
Scalability	The system should be able to handle **at least 1000 users at the same time** without slowing down.
Maintainability	The system should be designed in a way that makes it **easy to update, fix issues, and improve** without breaking the user interface.

Table 5.1: Non-Functional Requirements (NFRs) for Sprint 1

Chapter 6

Work Division

The tasks in this sprint were equally distributed among the three team members, ensuring an efficient and balanced workflow:

- **Haider Abbas** focused on the development of core pages, including the **login and signup pages**, ensuring the foundational functionality of the system.
- **Saif Shahzad** took charge of workflow management, overseeing the **Trello Scrum Board** and handling **documentation in LaTeX** to maintain project organization.
- **Hamd Ul Haq** played a supportive and collaborative role in the iterative process. He assisted Haider Abbas in developing core pages, worked on the **dashboard page**, and contributed to **Trello management and documentation** alongside Saif Shahzad.

This structured division of tasks allowed for **seamless collaboration**, ensuring that each aspect of the project was handled effectively while maintaining **agility in development**.

Chapter 7

Implementation screenshoots

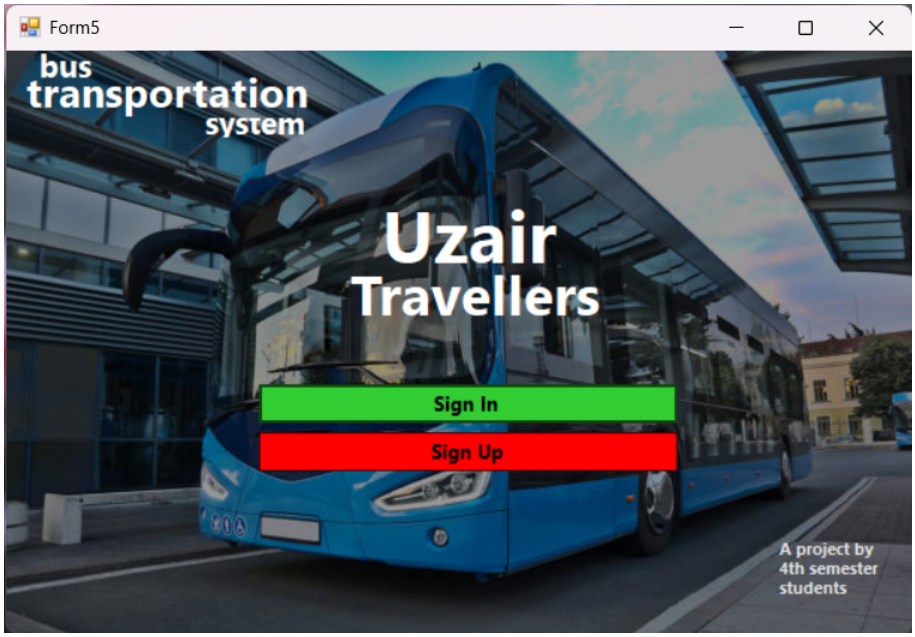

Figure 7.1: Landing Page

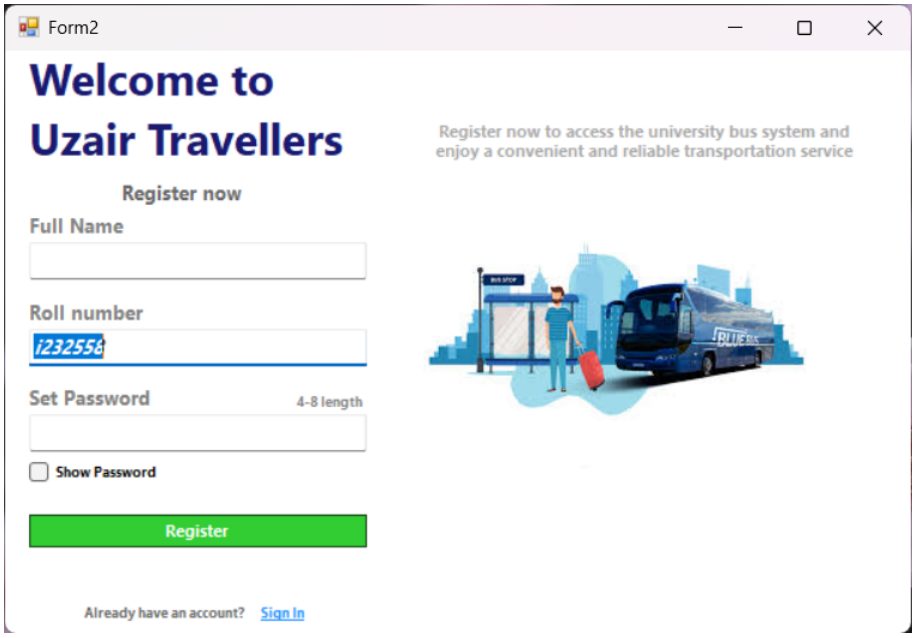

Figure 7.2: Sign up Page

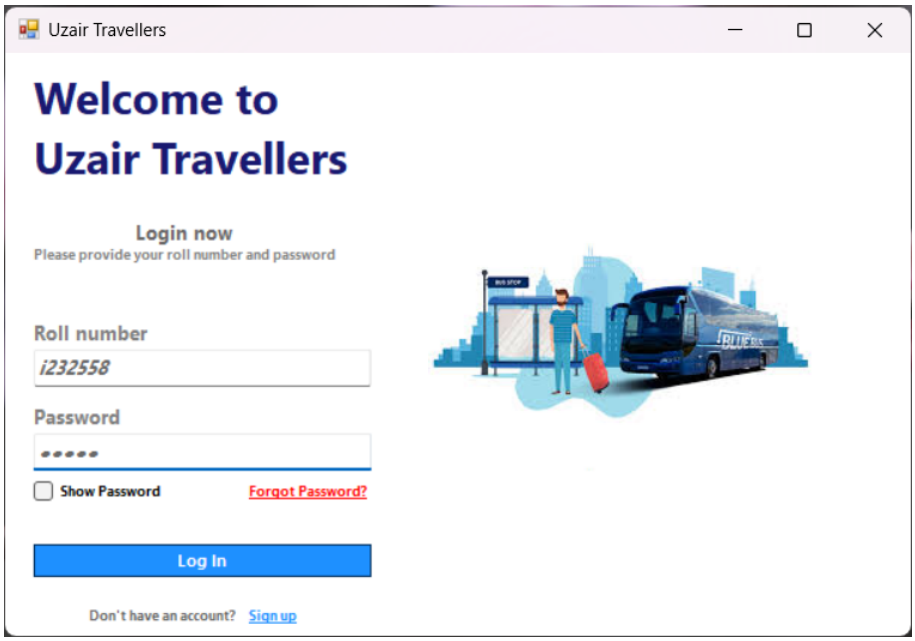

Figure 7.3: Sign In page

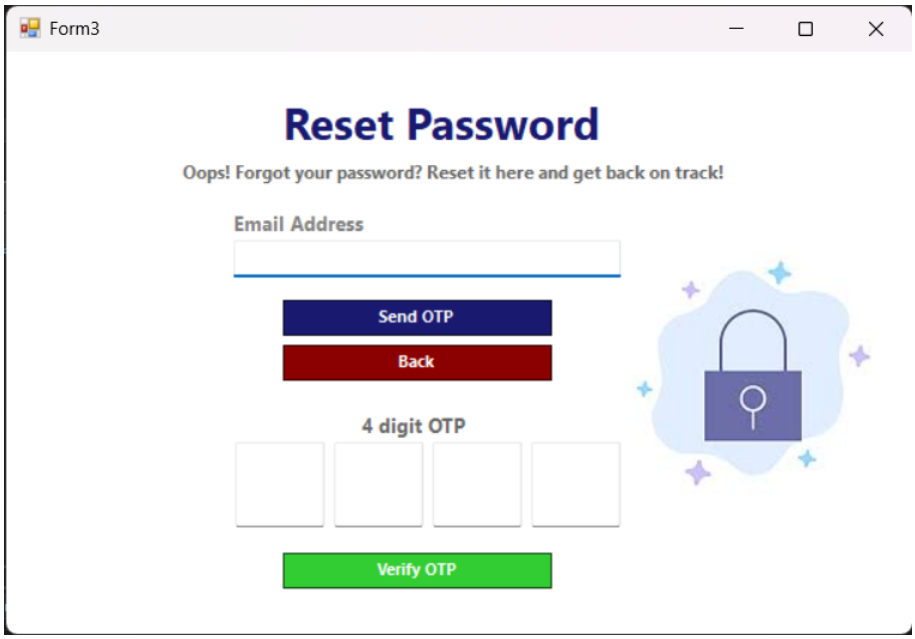

Figure 7.4: Reset Password confirmation Page

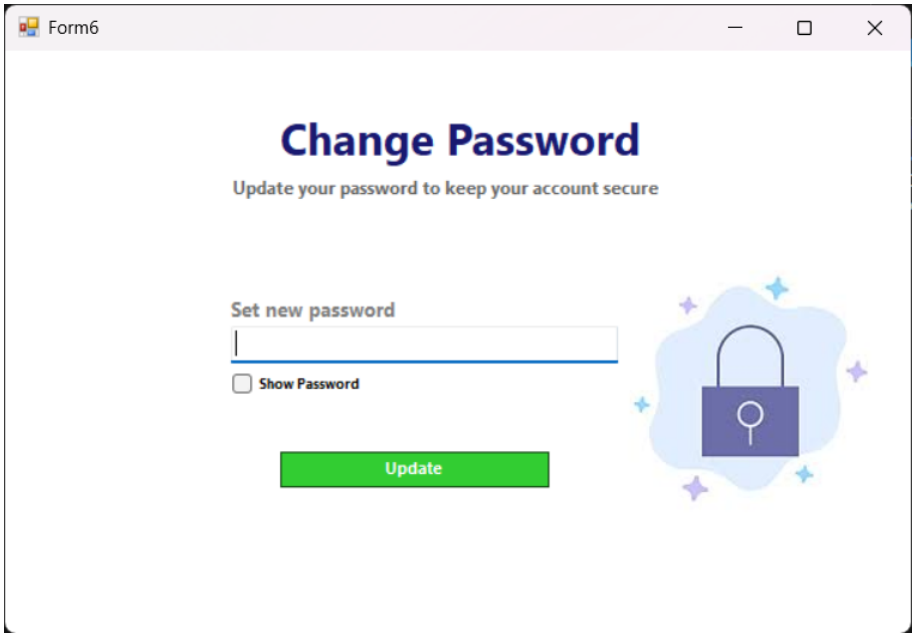

Figure 7.5: Change Password through email sent

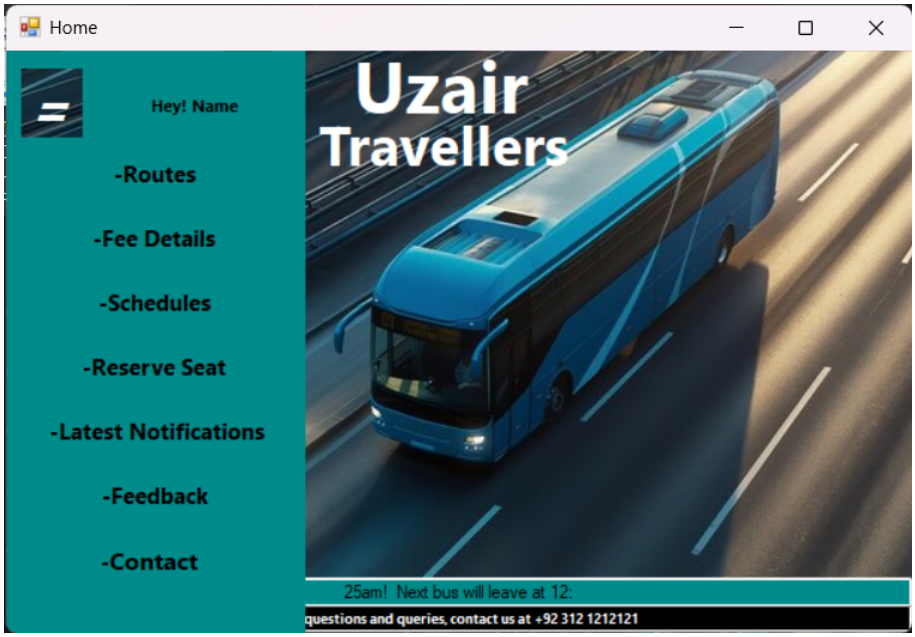

Figure 7.6: Dashboard UI