MAJOR PROJECT REPORT ON

Instant Dispatch

"Delivery Management System"

UNDERTAKEN AT

SANT SINGAJI INSTITUTE OF SCIENCE AND MANAGEMENT

BY

Nikita Yadav

SUBMITTED TO

VIKRAM UNIVERSITY UJJAIN (M.P.)

Partial fulfilment of the Requirements for the Award of the Degree Of

Bachelor of Computer

Application (B.C.A.)

Guide Name

Ms. Tanushree Soni

Sant Singaji Institute of Science and Management Sandalpur [Dewas]

Certificate

Reference No: ……. Date: ……

Project Completion Certificate

This is to certify that **Ms. Nikita Yadav** student of **BCA (Final Year)** of **Sant Singaji Institute of Science and Management**, has successfully completed the project work entitled **"Instant Dispatch - Delivery Management System"** under the guidance of **Ms. Tanushree Soni** is a bona fide piece of work carried out at **Sant Singaji Institute of Science and Management Sandalpur.**

The project entitled **"Instant Dispatch - Delivery Management System"** developed by **Nikita Yadav** in the **SSISM** and she has put at least 200 hours of laboratory work during the tenure of the project with the guide to complete this project. All the **prescribed certificates** are attached after the completion of all the formalities of the project work as pre-schedule, including internal examination.

Place **Signature of Principal**

Date **Signature of HOD**

Certificate of Attendance

Reference No: ……. Date: ……

This is to certify that **Ms. Nikita Yadav,** Student of BCA **(Final Year)** of **Sant Singaji Institute of Science and Management**, has successfully completed the project work entitled **"Instant Dispatch - Delivery Management System"** and he has put at least 200 hours of laboratory work during the stipulated period of the project at **Sant Singaji Institute of Science and Management, Sandalpur.**

Place **Signature of Guide**

Date **Signature of Principal**

Declaration

I'm **Nikita Yadav** of **Sant Singaji Institute of Science & Management** declarethat the project report submitted by me under the guidance of **Ms. Tanushree Soni** is a bona fide work for the partial fulfilment of the requirement of the **BCA Final Year project work**. I have incorporated all the suggestions provided by my guide time to time.

I further declare that to the best of my knowledge this project contains my original work & does not contain any part of any work which has been submittedfor the award of any degree either in this university or in any other university/Deemed university/Institute etc. Without proper citation and I shall befully responsible for any plagiarism found at any stage.

Name & Signature of the Guide **Name & Signature of the Student**

Project Approval Certificate

Reference No: ……. Date: ……

This to certify that **Ms. Nikita Yadav, a** student of **BCA (Final Year)** of **Sant Singaji Institute of Science & Management**, has successfully completed the project entitled **"Instant Dispatch - Delivery Management System"** under my guidance. I have regularly assessed the progress of the work and suggested the correction whenever required. The student has incorporated all the suggestions provided by me in this project. This project is bona fide piece of work of the standard of BCA project work carried out by the student under my supervision. Internal examination has been completed in my presence and student's performance was satisfactory and hence this project is approved for the submission and valuation thereof.

Signature of Guide

Place: **Signature of Principal**

Date: **Seal of the Institute**

Table of Contents

Introduction

✓ Project Overview

At **"Instant Dispatch"**, we understand the everyday challenges of finding reliable and fast delivery services for your parcels and packages. That's why we created this platform – to put an end to the hassle and save you precious time.

Imagine the countless hours spent searching for a reliable delivery service or a trustworthy rider. With Instant Dispatch, those challenges become a thing of the past. We're here to simplify your delivery experience, making it effortless to find the perfect delivery solution for your needs.

No more wasting time scrolling through endless options or dealing with uncertainty. Instant Dispatch is your solution to quick, reliable, and efficient delivery services. We're dedicated to solving the problems that have been a roadblock in your delivery journey, ensuring that your time is spent on what matters most – getting your packages delivered safely and on time.

Let **"Instant Dispatch"** be the key to unlocking a stress-free and time-saving approach to package delivery. Your convenience is our top priority, and we're here to make every delivery a breeze. Welcome to a world where your delivery worries are a thing of the past!

User Phase

1. Customer Phase

Discover your ideal delivery service effortlessly with Instant Dispatch's Customer Module. Browse through a diverse range of delivery options, review rider ratings to make informed choices, and confidently book the perfect delivery service for your needs. Your delivery starts with a few simple taps, ensuring a seamless and stress-free experience.

2. Rider Phase

Embark on a flexible and rewarding journey as a rider with Instant Dispatch. Register effortlessly, set your availability preferences, and accept delivery bookings that align with your schedule. Connect with customers who value your services in this user-friendly delivery network, making each delivery a positive and fulfilling experience.

3. Admin Phase

The Admin Module within Instant Dispatch is the backbone of our platform, ensuring a secure and reliable environment. Administrators wield the authority to verify and accept rider registrations, guaranteeing the quality and trustworthiness of our community. With the power to swiftly manage users, orders, and riders, admins maintain order and safety, responding promptly to reported issues or violations. The Admin Module is our guardian, upholding standards, verifying credentials, and fostering a secure space for a trustworthy and enjoyable delivery experience.

1.2 Problems and solutions of the project

1.2.1 Existing System

Individuals often face considerable challenges when searching for reliable delivery services to fulfill their shipping needs. The existing options lack a streamlined and trustworthy platform, making it time-consuming and sometimes frustrating to find a suitable delivery service. Users may encounter difficulties in assessing the reliability of both delivery services and riders, leading to concerns about the overall safety and quality of their delivery experiences. This lack of a centralized and efficient solution can result in wasted time, inconvenience, and a diminished sense of confidence when sending packages.

1.2.2 Proposed System

"Instant Dispatch" addresses this prevalent issue by providing a comprehensive platform that streamlines the process of finding the ideal delivery service and rider. By offering a user-friendly interface, detailed profiles, and a transparent rating system, Instant Dispatch ensures that users can confidently select a reliable delivery service that meets their specific requirements. The platform's commitment to safety, quality, and efficiency aims to eliminate the uncertainties associated with traditional methods of finding delivery services, offering a one-stop solution that saves time and provides peace of mind for every delivery.

Feasibility Report

Instant Dispatch's initial investigation evaluates its overall feasibility, emphasizing Technical, Operational, and Economic viability. The study assesses the potential of introducing new modules and debugging the existing system. Key aspects covered include: of the preliminary investigation:

✓ Technical Feasibility

✓ Operation Feasibility

✓ Economic Feasibility

2.1. Technical Feasibility

The technical issue usually raised during the feasibility stage includes:

- Does the necessary technology exist to do what is suggested?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Technical feasibility is paramount for Instant Dispatch, requiring a meticulous evaluation of its platform's robustness. This involves assessing scalability, compatibility across diverse devices, and seamless integration with various operating systems. Emphasis is placed on:

- Maintaining a stable server infrastructure
- Efficient data management
- Real-time tracking capabilities using Socket.IO
- Ability to adapt swiftly to emerging technologies

2.2. Operational Feasibility

Operational feasibility is at the forefront of Instant Dispatch's considerations, examining how seamlessly the platform integrates into existing processes. The focus is on:

- Minimizing disruptions during the transition
- Ensuring a smooth adoption process for users
- Assessing user adoption and training needs
- Understanding potential impacts on day-to-day operations

and implemented?

2.3. Economic Feasibility

Economic feasibility forms the bedrock of Instant Dispatch's strategic planning, involving a thorough cost-benefit analysis. This comprehensive examination weighs:

- Development and operational expenses
- Potential revenue streams
- User acquisition costs
- Pricing models
- Market demand analysis

System Analysis

Software Requirement Specification Overview

The need for **"Instant Dispatch"** arises from the prevalent challenges individuals face in finding reliable delivery services. With the increasing demand for convenient and trustworthy delivery solutions, there's a clear gap in the market for a platform that seamlessly connects users with reliable riders. **Instant Dispatch** addresses this need by providing a user-friendly interface where users can easily browse, book, and track deliveries, ensuring a hassle-free delivery experience.

○ Home module

Through the Home module, users can access various functionalities including:

- Browse available delivery services
- View rider profiles and ratings
- Access booking history
- Track active deliveries in real-time

○ Customer Module

In the Customer Module of Instant Dispatch, users can perform a variety of tasks:

- Browse through available riders and delivery options
- View detailed profiles of riders including ratings and reviews
- Specify delivery requirements (pickup address, delivery address, package details)
- Book delivery services
- Track deliveries in real-time
- View order history
- Receive notifications about delivery status

- **Customer Functionalities:**

  - Sign In or Sign Up
  - Book delivery services
  - Track orders in real-time
  - View order history
  - Rate and review riders
  - Manage profile

- **Rider Module**

  In the Rider Module of Instant Dispatch, riders can:

  - Manage their availability
  - View booking requests
  - Accept or reject delivery requests
  - Update their profiles
  - Navigate to pickup and delivery locations
  - Update delivery status
  - View earnings and performance metrics

- **Rider Functionalities:**

  - Register and create profile
  - Set availability status
  - Accept/reject delivery requests
  - Update delivery status
  - View earnings
  - Access navigation tools

- ## Admin Module

In the Admin Module of Instant Dispatch, administrators have comprehensive control:

- Verify and approve rider registrations
- Monitor platform performance and analytics
- Manage users, riders, and orders
- Block or unblock users as needed
- View all orders and their status
- Generate reports and analytics

- ## Admin Functionalities:

- Admin login
- Manage users (block/unblock)
- Manage riders (approve/reject, activate/deactivate)
- View all orders
- Monitor platform analytics
- Handle reported issues

System Requirement Analysis

4.1 Hardware Requirements

- Android mobile device, Computer, Laptop is needed.
- 3 GB RAM and above.
- 32 ROM and above.

4.2 Software Requirements

- Web Browser is needed.
- Operating System Compatibility.
- Internet Connection.

System Design

5.1 Module Design

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for **Instant Dispatch** or any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities - design, code and test that is required to build and verify software.

The significance of quality is paramount in Instant Dispatch. Design serves as the cornerstone where quality is cultivated within software development. It offers us visualizations of the platform, enabling assessment of Instant Dispatch's integrity. Design is indispensable for accurately translating user perspectives into the final product. It lays the groundwork for all subsequent software engineering processes. Without a robust design, there is a risk of constructing an unreliable system, challenging to evaluate until the final stages of development.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

5.2 Data Flow Diagrams

Data Flow Diagrams (DFDs) are graphical tools utilized to illustrate and analyze the flow of data within the Instant Dispatch system. They serve as the central tool and foundation upon which other system components are constructed. The transformation of data from input to output, passing through various processes, is logically described independently of the system's physical components, forming the basis of logical data flow diagrams. Physical data flow diagrams depict the actual implementation and movement of data between users, departments, and workstations.

A comprehensive system description consists of a series of data flow diagrams, developed using familiar notations such as Yourdon, Gane, and Sarson. Each component in a DFD is labeled with a descriptive name, with processes further identified by a number for identification purposes within the Instant Dispatch platform. DFDs are developed across multiple levels, with each process in lower-level diagrams broken down into more detailed DFDs in subsequent levels. The top-level diagram, known as the context diagram, features a single process that is crucial for understanding the current system. Processes in the context level diagram are expanded into additional processes in the first-level DFD.

DFD Symbols

In the DFD, there are four symbols

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow. It is the pipeline through which the

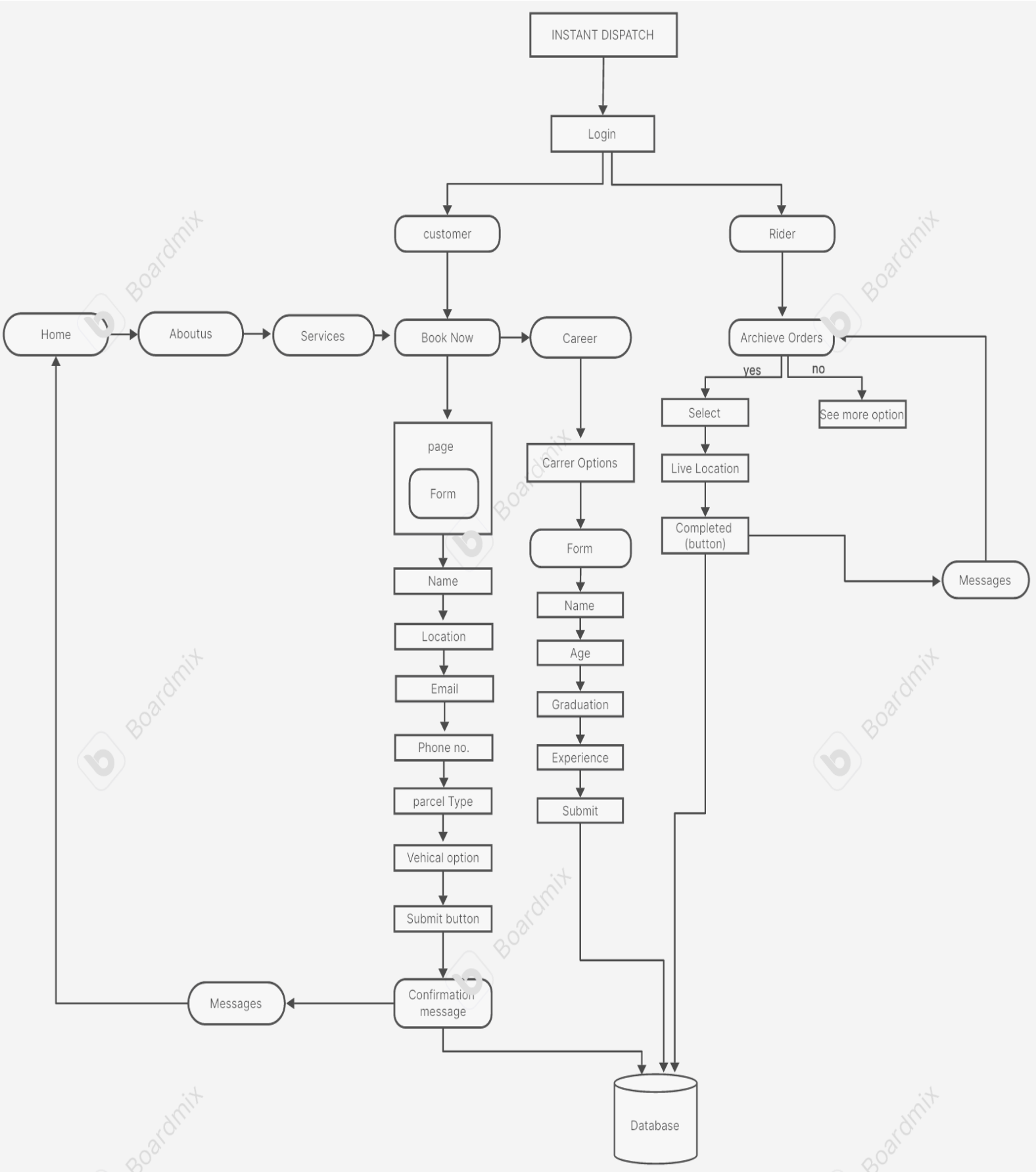
information flows.

3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.

Data Flow

1. A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4. A data flow to a data store means update (delete or change).
5. A data flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.


INSTANT DISPATCH
Login
customer
Rider
Home
Aboutus
Services
Book Now
Career
Archieve Orders
yes
no
Select
See more option
page
Form
Carrer Options
Live Location
Completed (button)
Form
Messages
Name
Name
Location
Age
Email
Graduation
Phone no.
Experience
parcel Type
Submit
Vehical option
Submit button
Confirmation message
Messages
Database

MongoDB Database

MongoDB is a modern, flexible, and open-source NoSQL database system. MongoDB stores data in flexible, JSON-like documents, making it ideal for handling unstructured data. MongoDB's flexible schema and dynamic querying capabilities allow for agile development and scalability. MongoDB is widely adopted for its high performance, scalability, and ease of use in modern application development.

Foreign Key

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

Aggregate queries

MongoDB's aggregate function allows you to perform complex operations on your data, such as grouping, filtering, and projecting, in a single operation. This enables you to extract valuable insights from your data efficiently and effectively.

1. Grouping documents based on specified criteria.
2. Aggregating data across multiple collections.
3. Filtering documents based on specific conditions.
4. Projecting fields to include or exclude in the output.
5. Sorting results based on one or more fields.
6. Performing mathematical and statistical calculations on the data.

System Testing and Implementation

6.1. Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for Instant Dispatch software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both large and small-scale systems.

6.2. Software Testing

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints, and validation criteria for **Instant Dispatch** software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for **Instant Dispatch** software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Taking another turn outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested

6.3. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. White Box Testing

This type of testing ensures that:

- All independent paths have been exercised at least once

- All logical decisions have been exercised on their true and false side

- All loops are executed at their boundaries and within their operational bounds

2. Basic Path Testing

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for **Instant Dispatch** all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph

Determine the Cyclometric complexity of resultant flow graph, using formula: V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number of Regions
Where V(G) is Cyclomatic complexity,E is
the number of edges,
N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. Conditional Testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

4. Data Flow Testing

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable was declared. The definition-use chain method was used in this typeof testing. These were particularly useful in nested statements.

5. Loop Testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for **"Instant Dispatch"** all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- Unstructured loops were resolved into nested loops or concatenated loops andtested as above.
- Each unit has been separately tested by the development team itself and all theinput have been validated.

User Module Output Screen

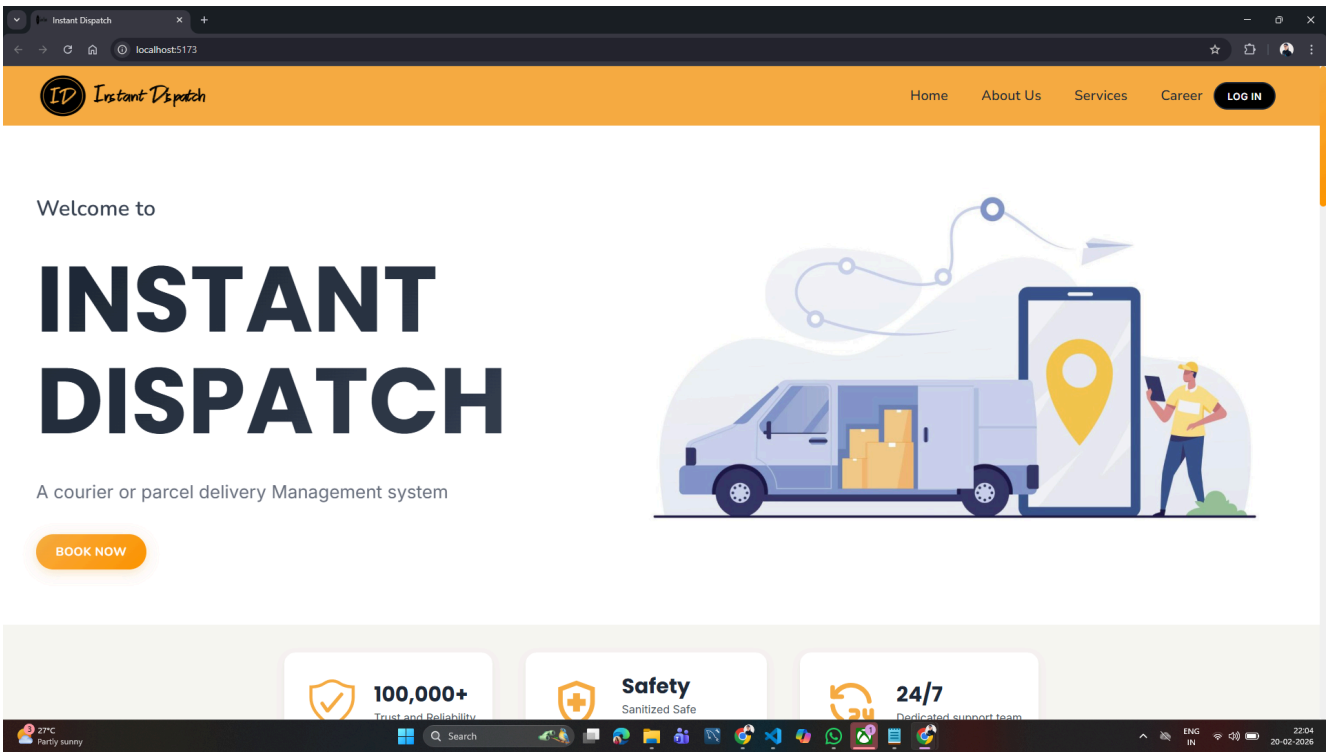


Fig. Landing Screen

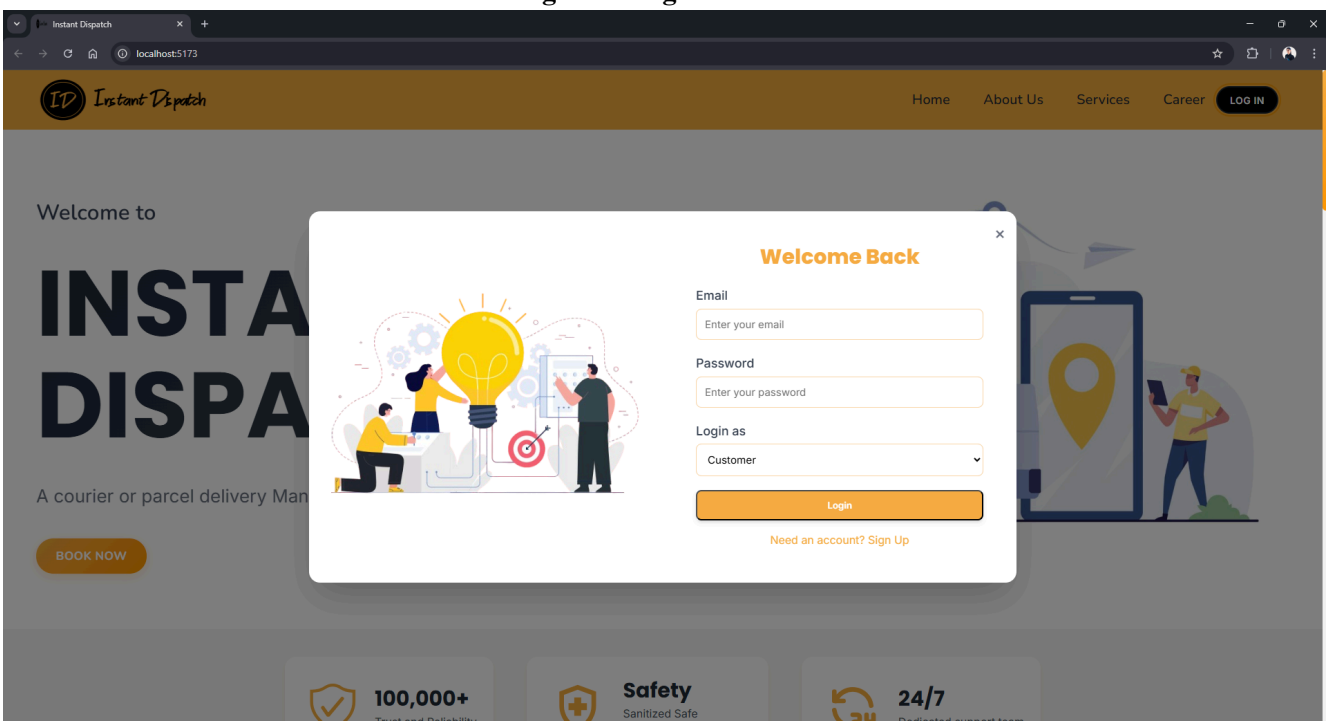
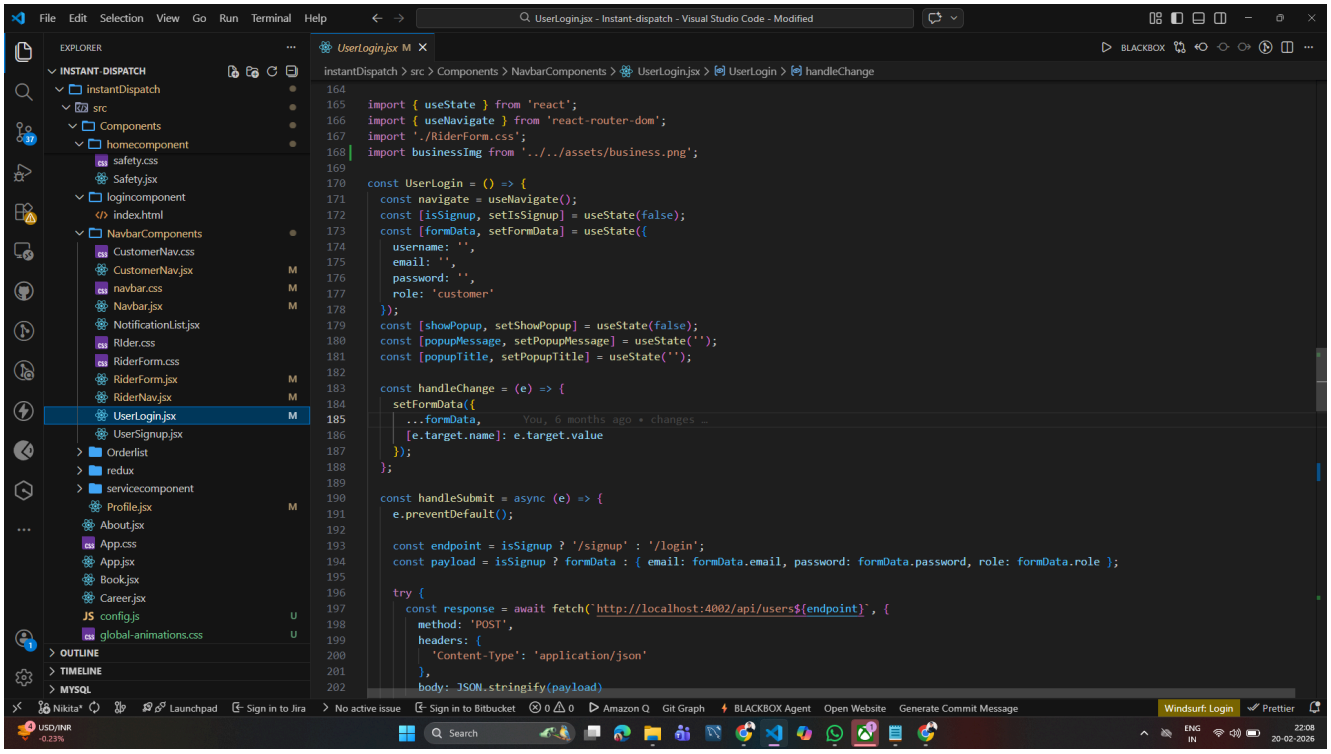


Fig. Login Screen



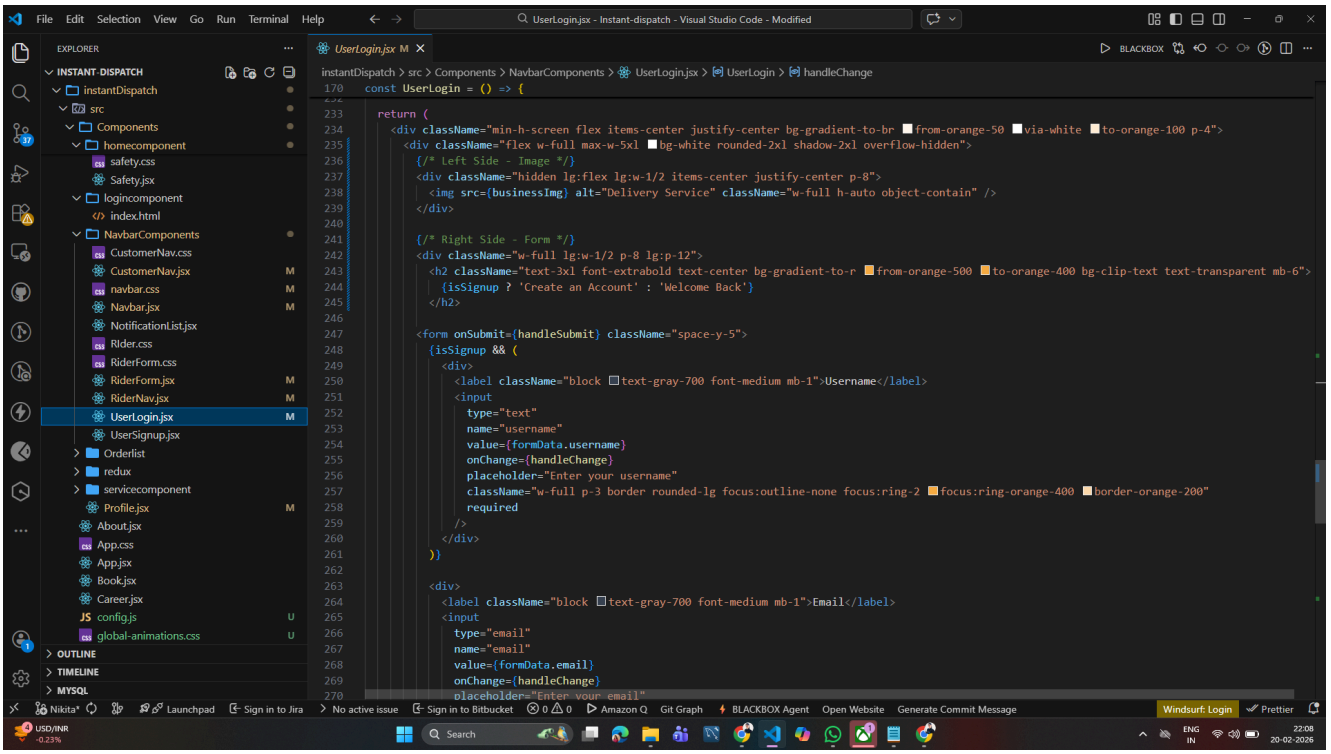


Fig. Login Screen Code

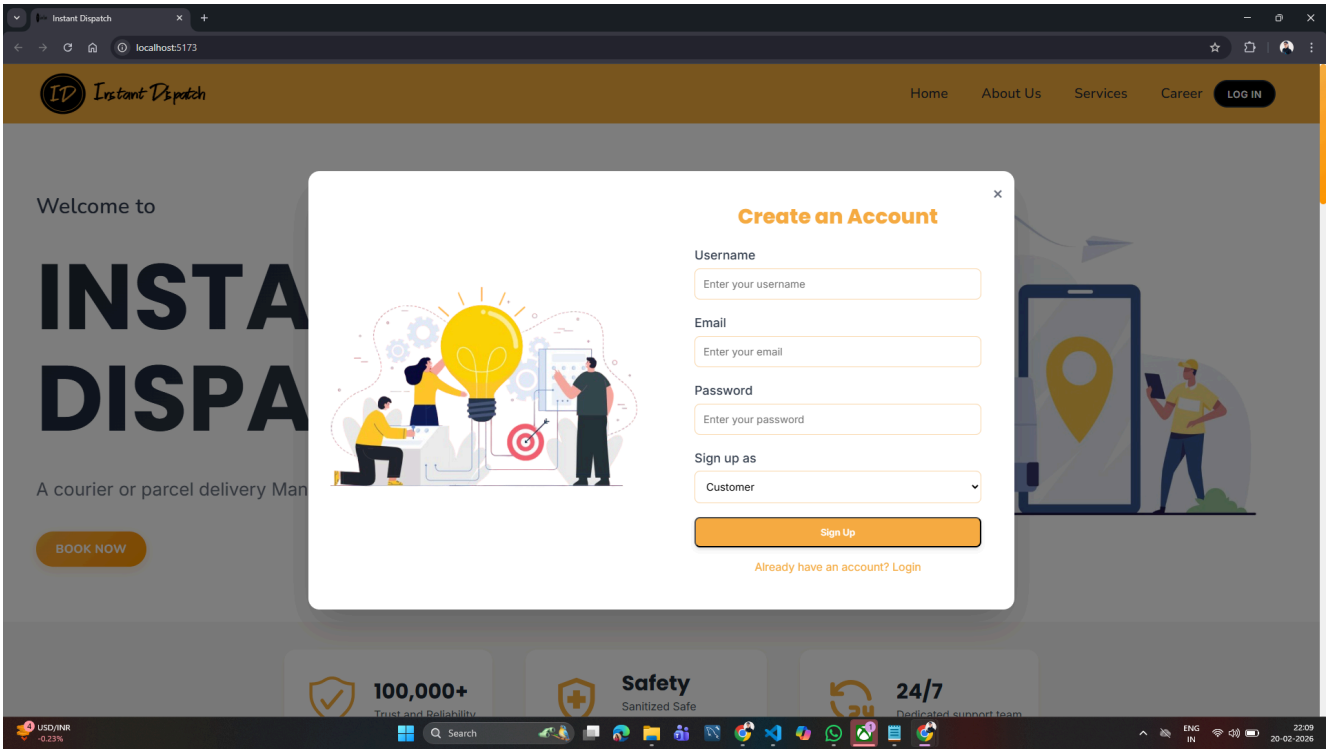
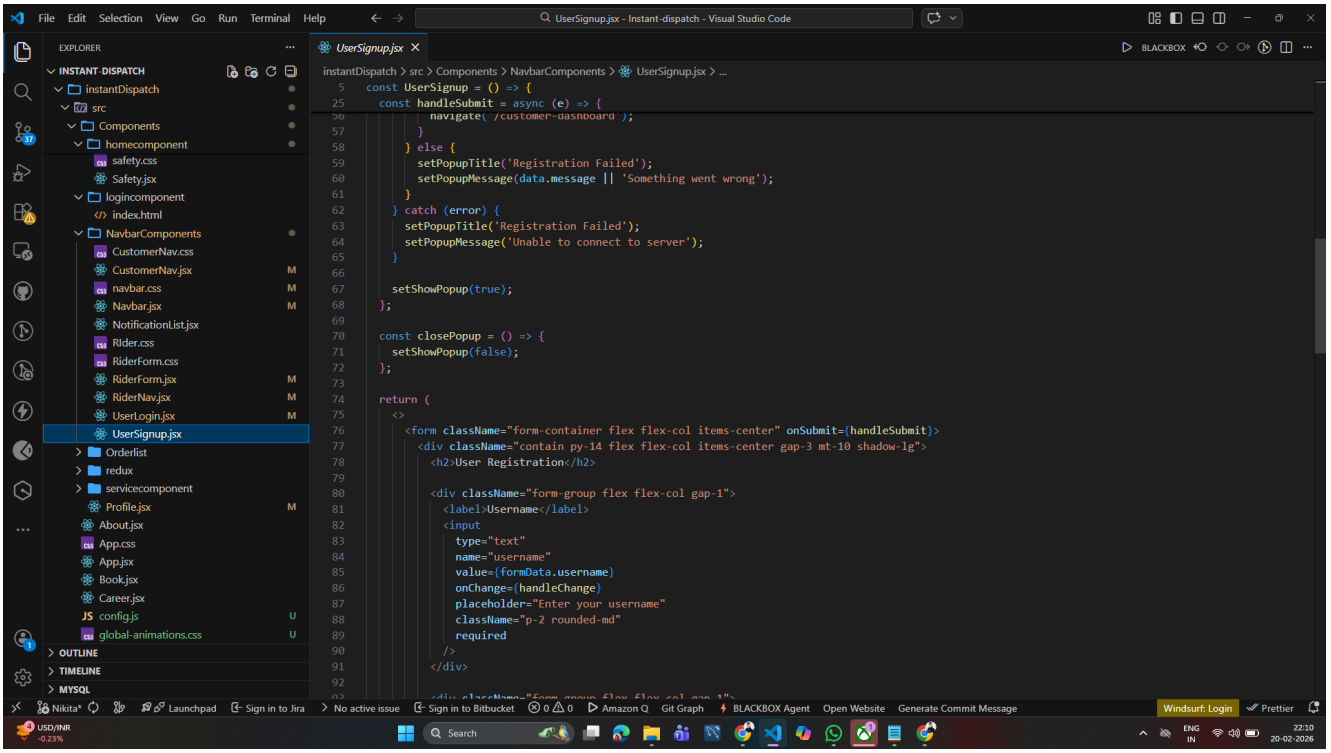


Fig. Registration Screen



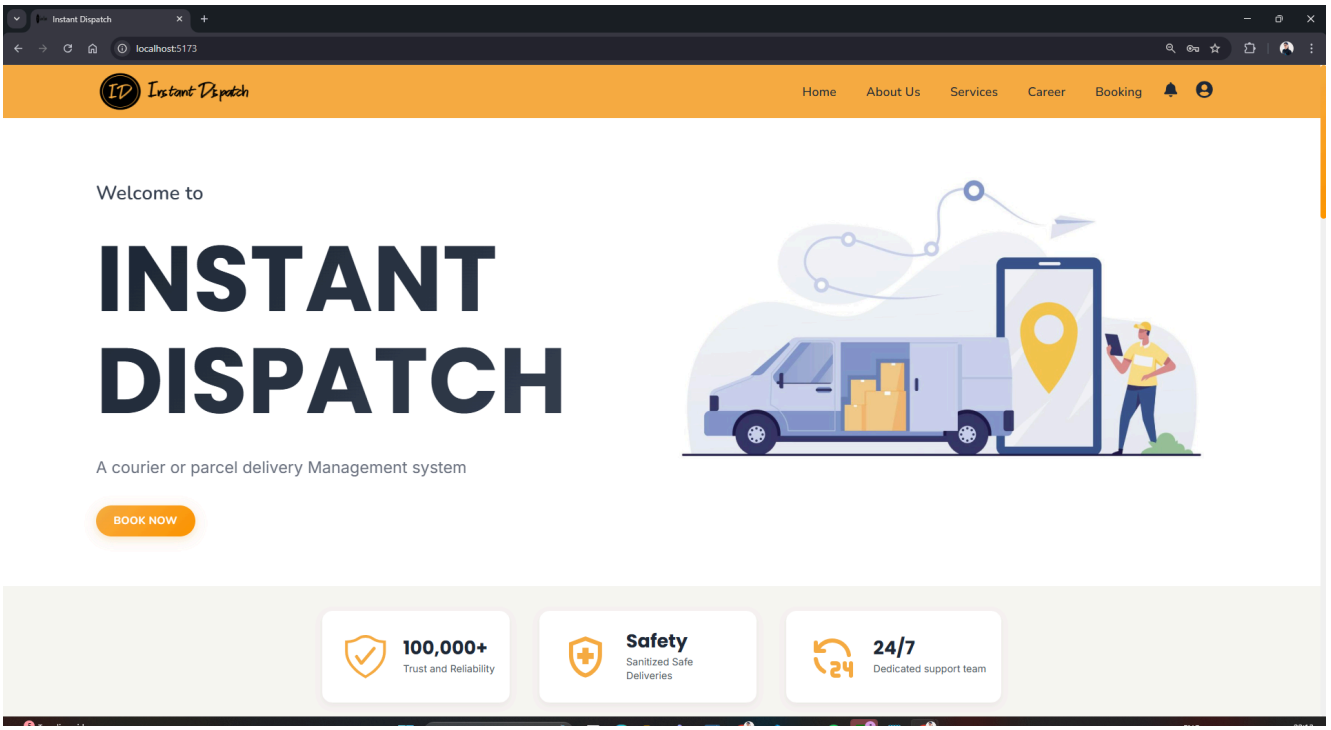


Fig. Overview Screen

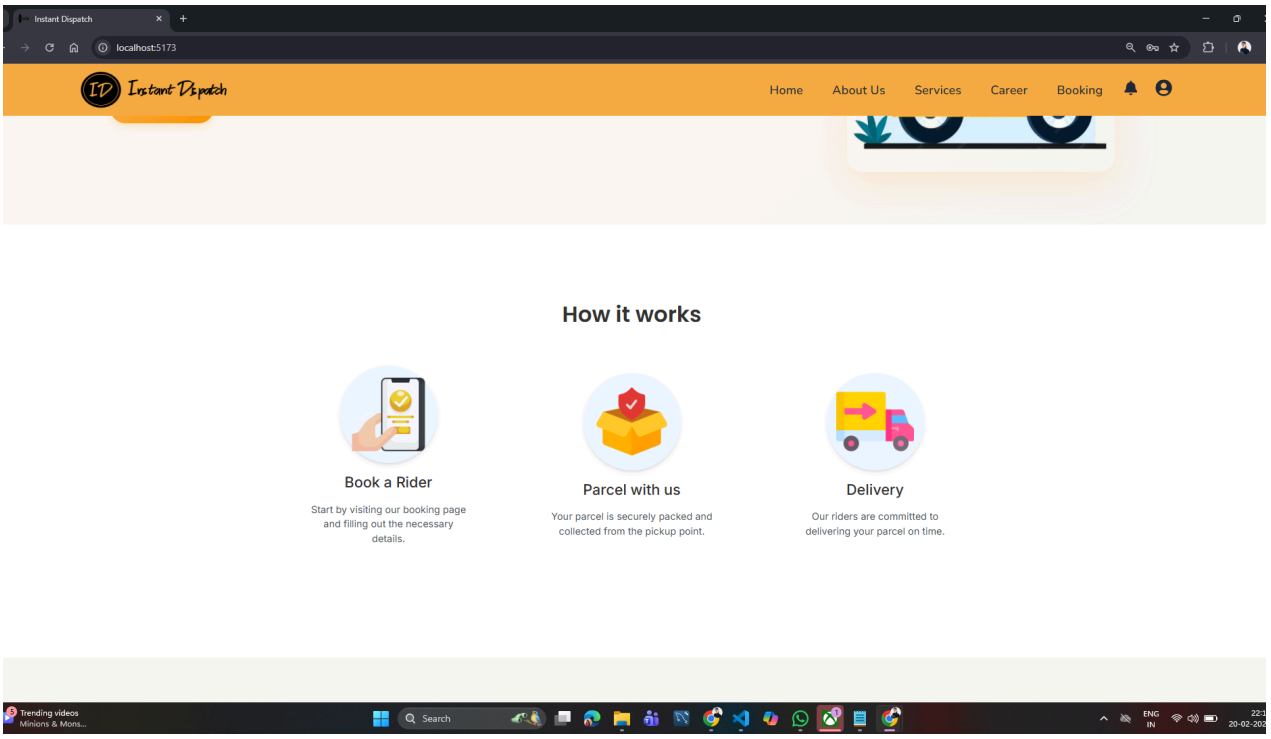


Fig. Overview Screen Code

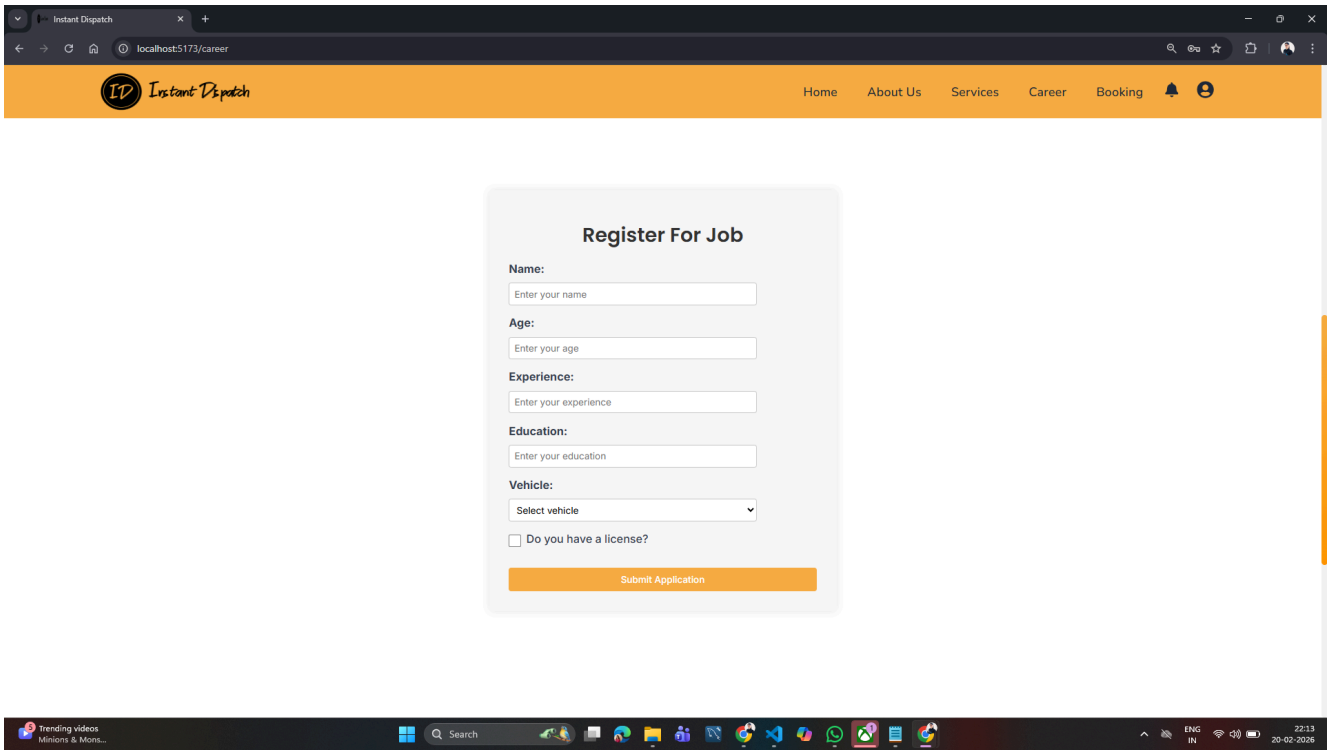


Fig. Career Form page

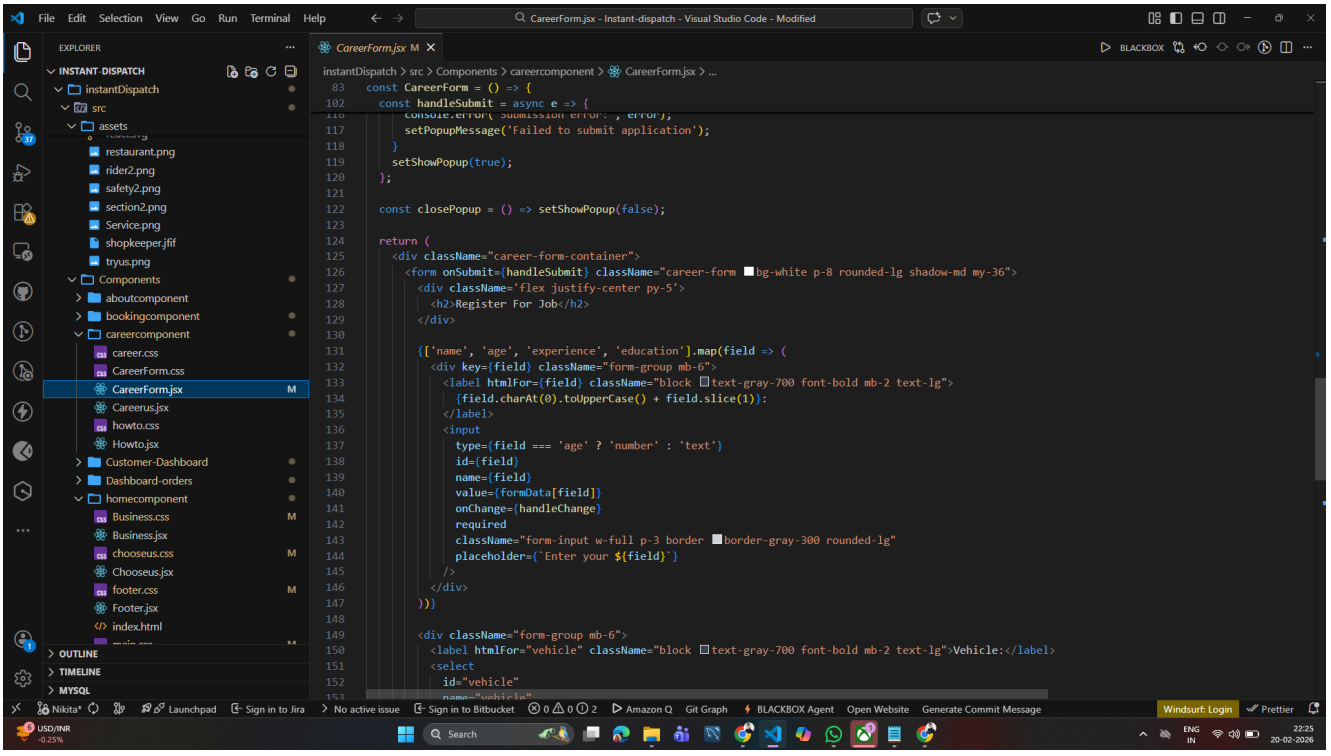


Fig. Career Form code

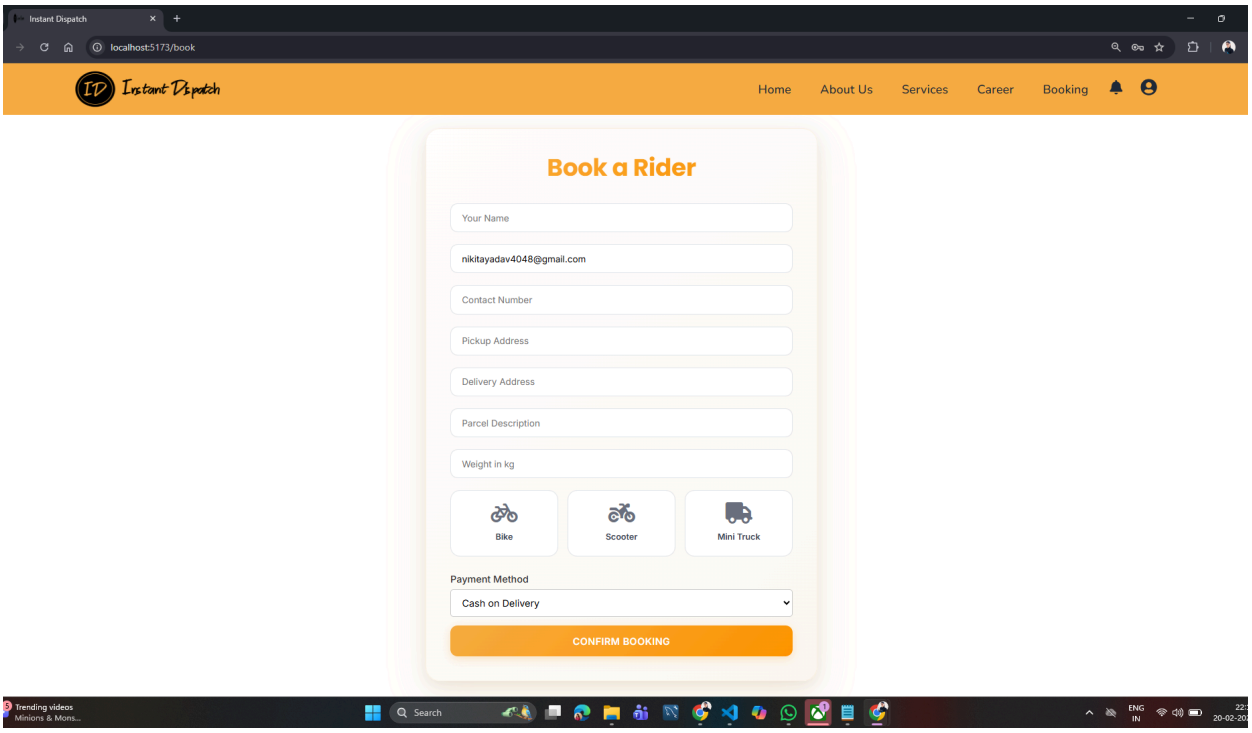


Fig. Booking Form Code

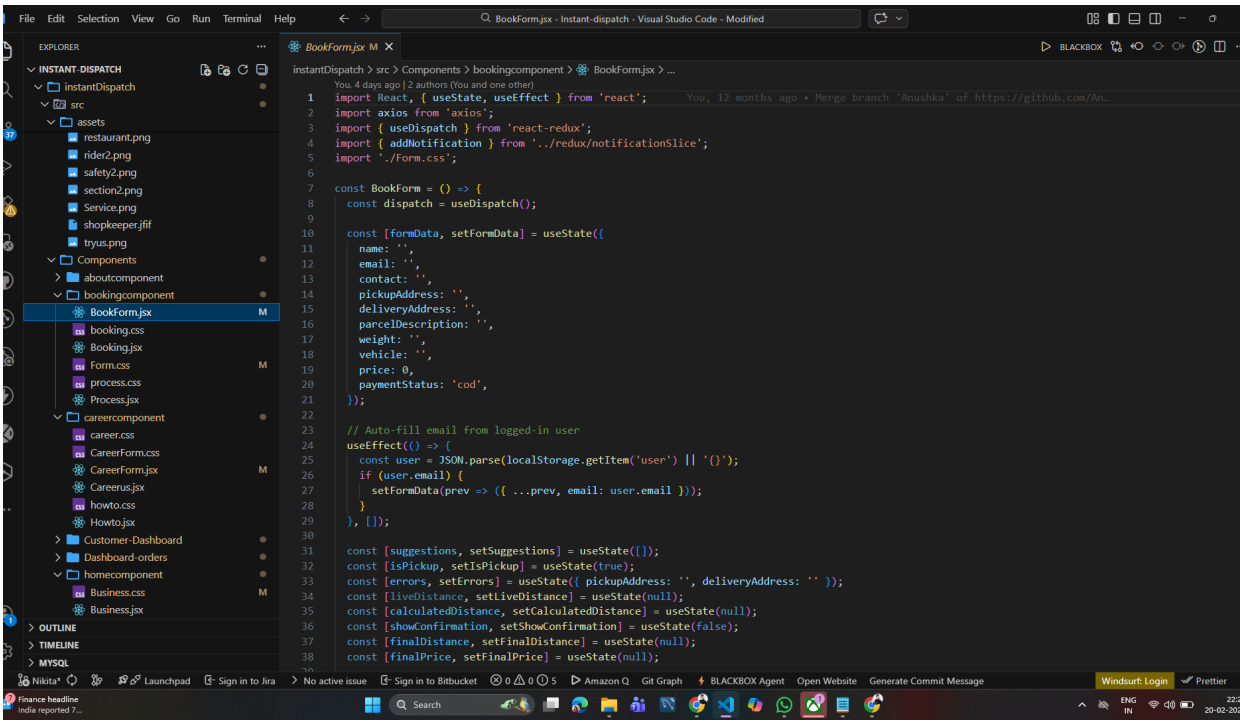


Fig. Booking Form Code

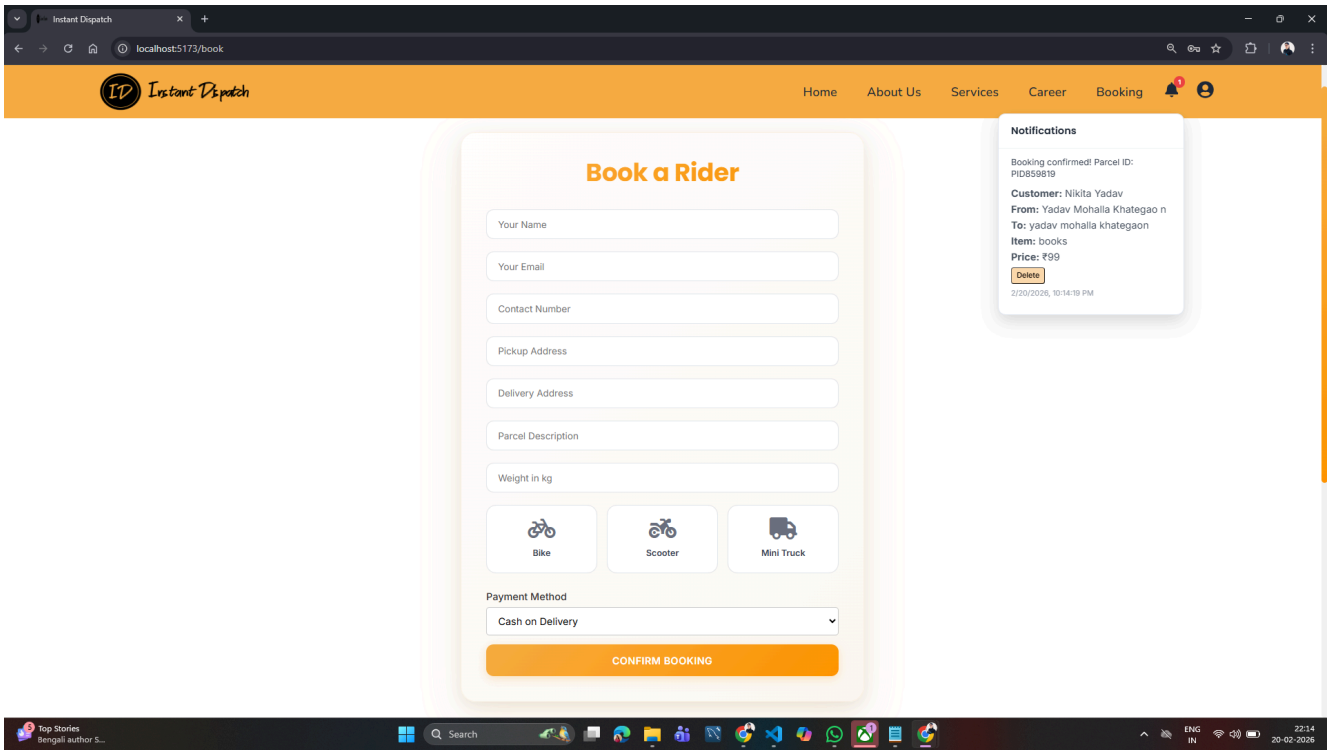


Fig. Notification section

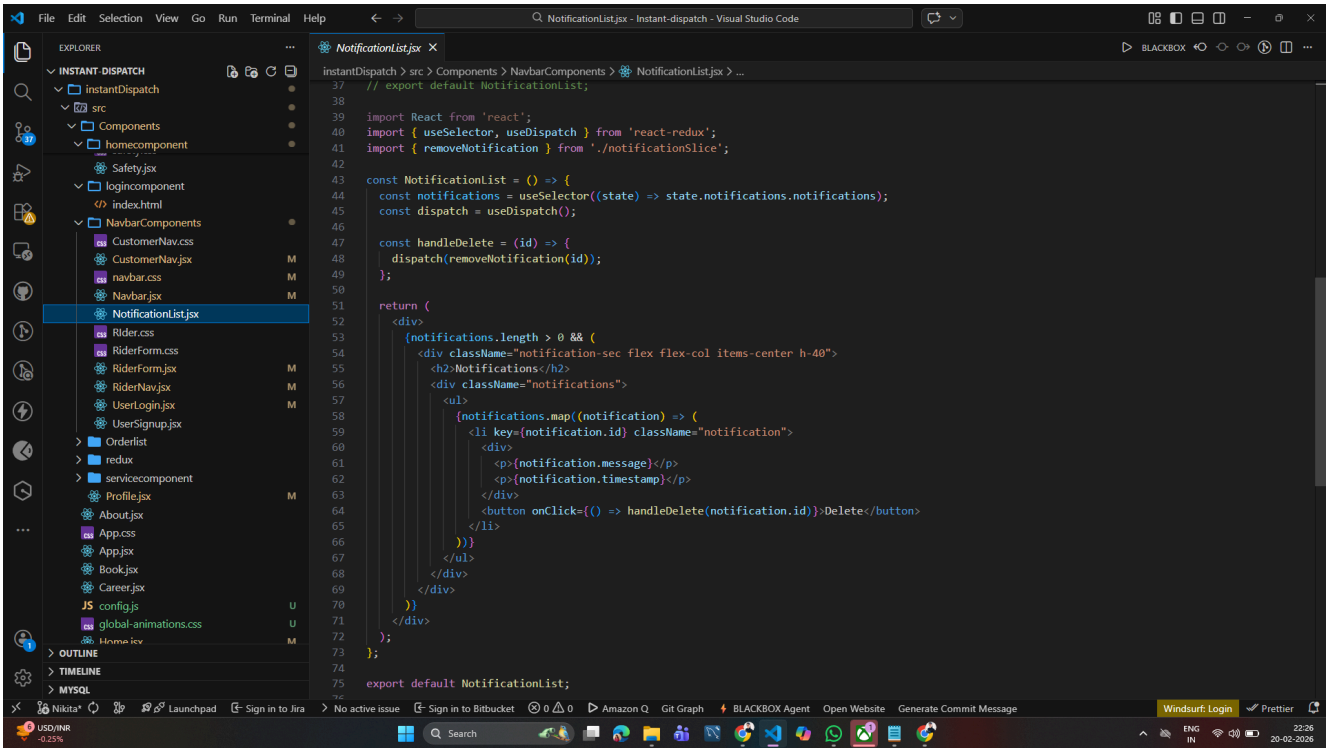


Fig.Notification code

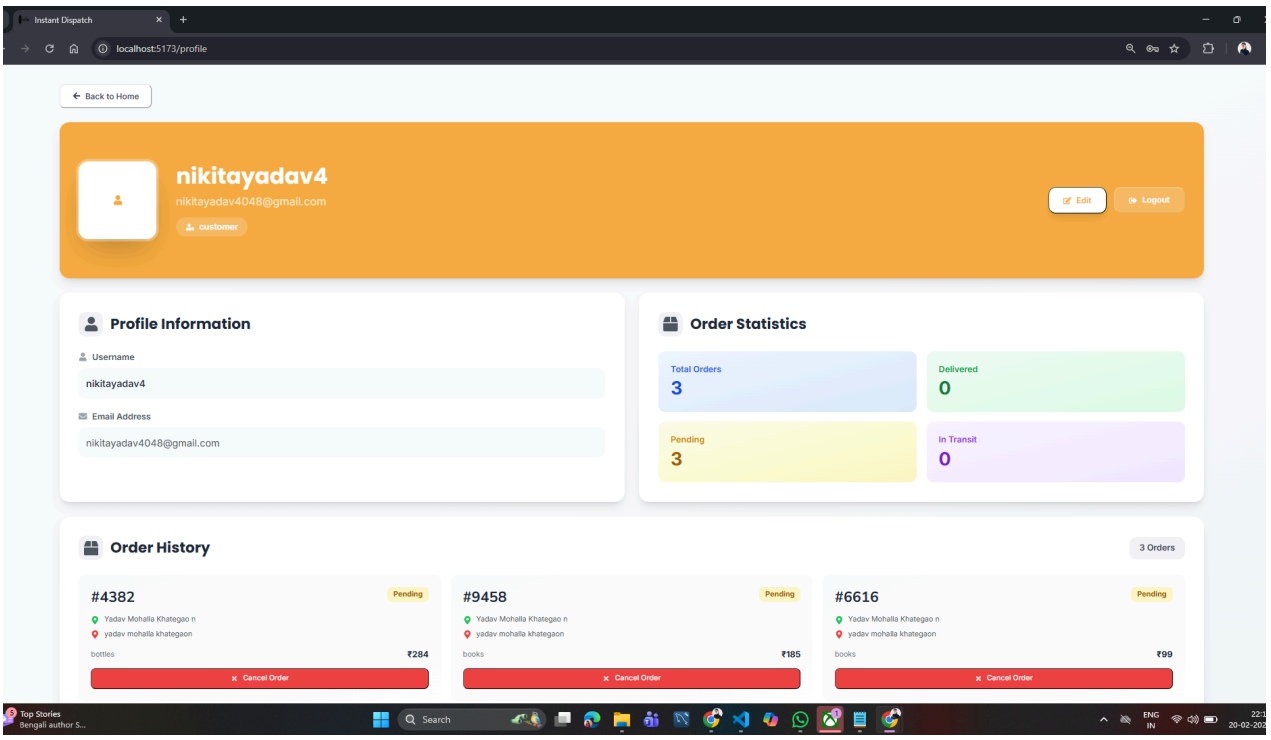


Fig. Profile screen code

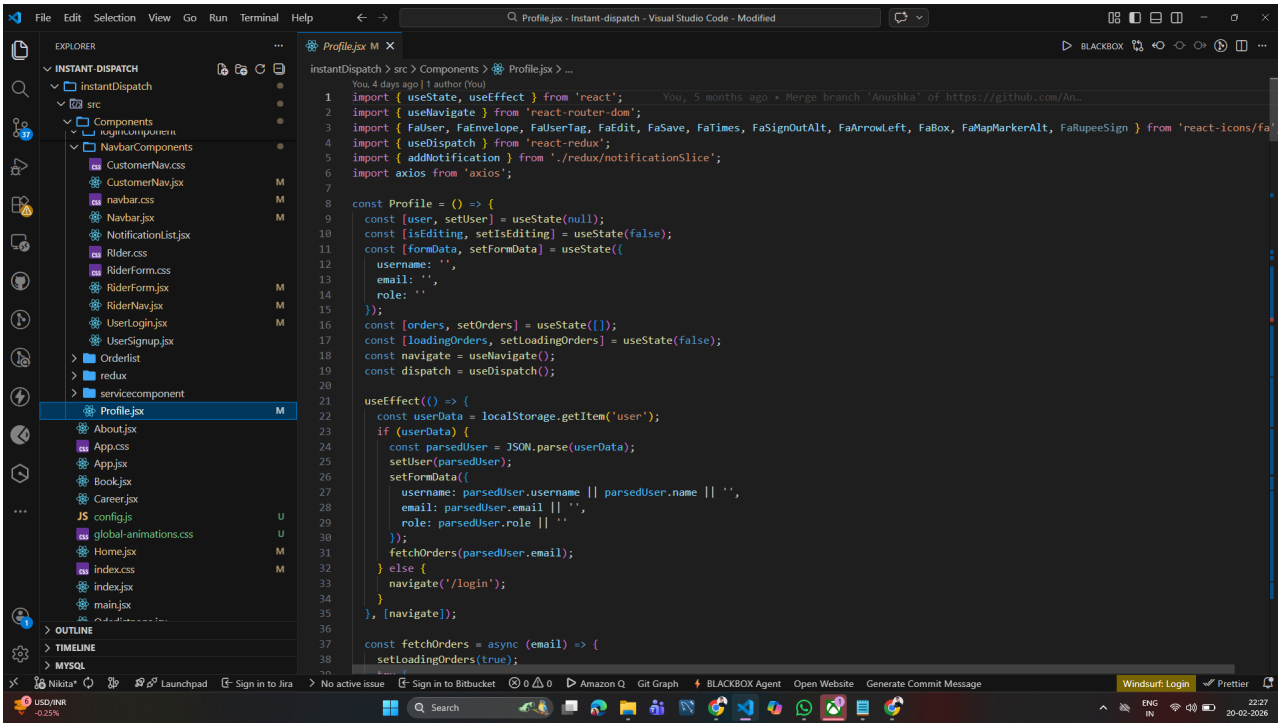


Fig.Profile section code

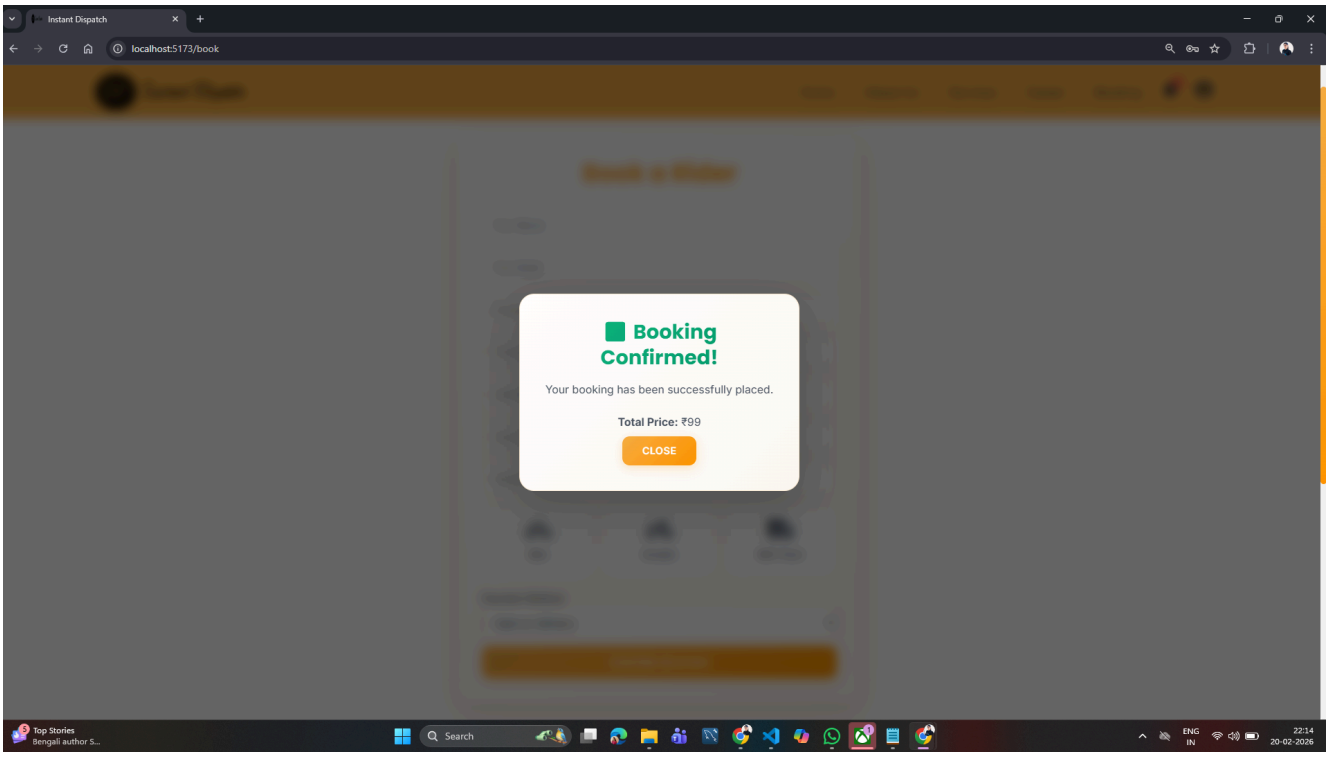


Fig. Booking Form confirmed

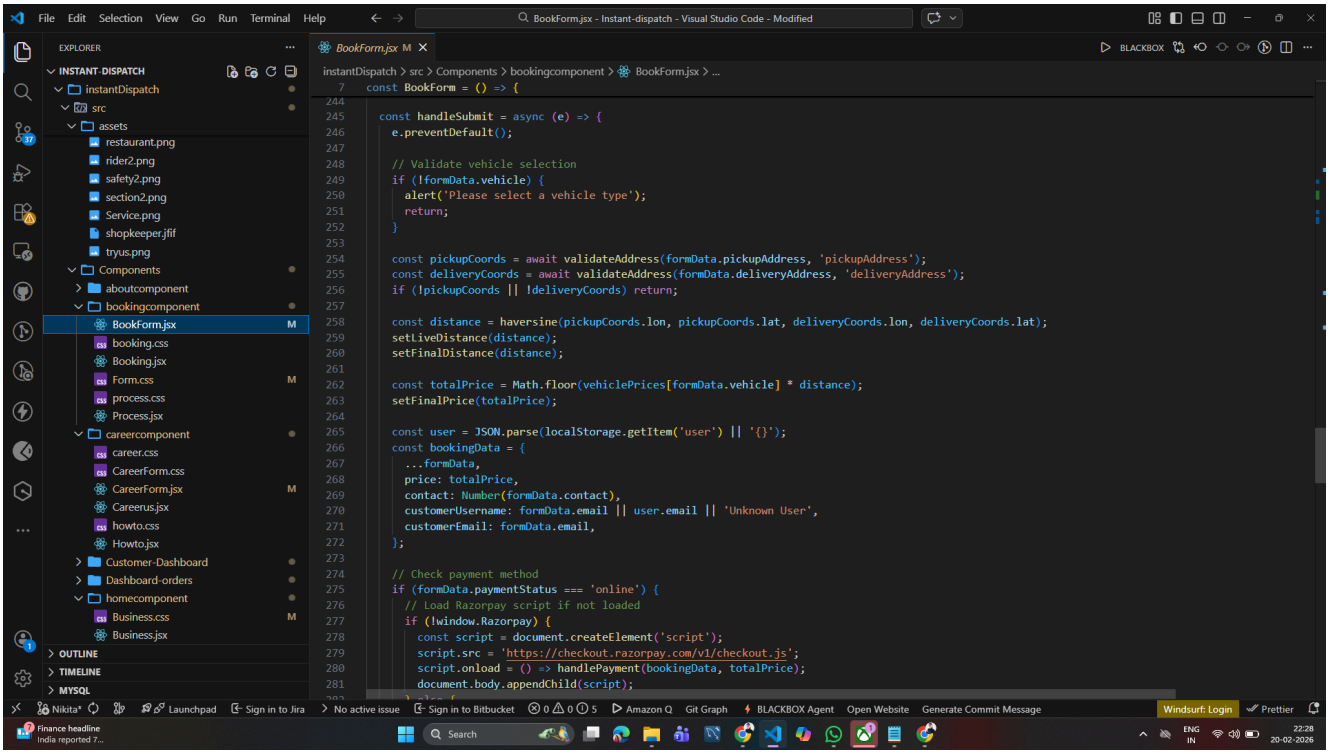


Fig.Booking Form confirmed code

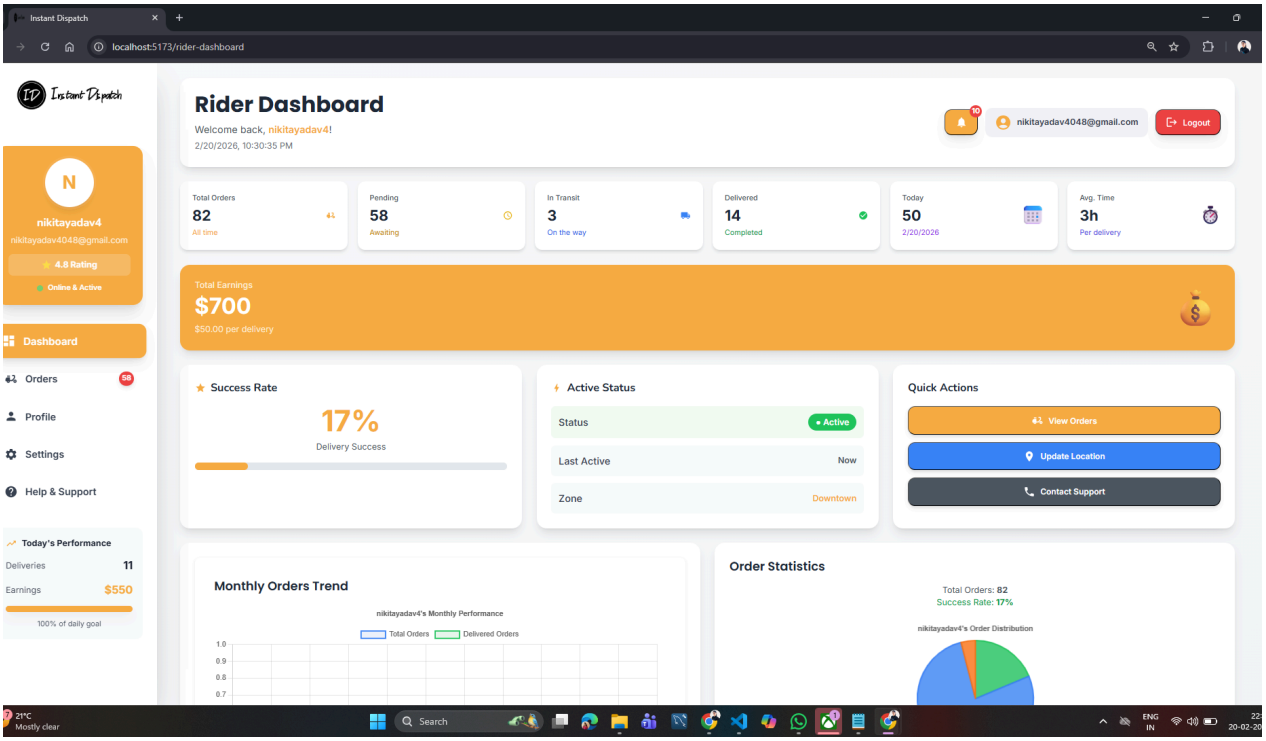


Fig. Rider Dashboard

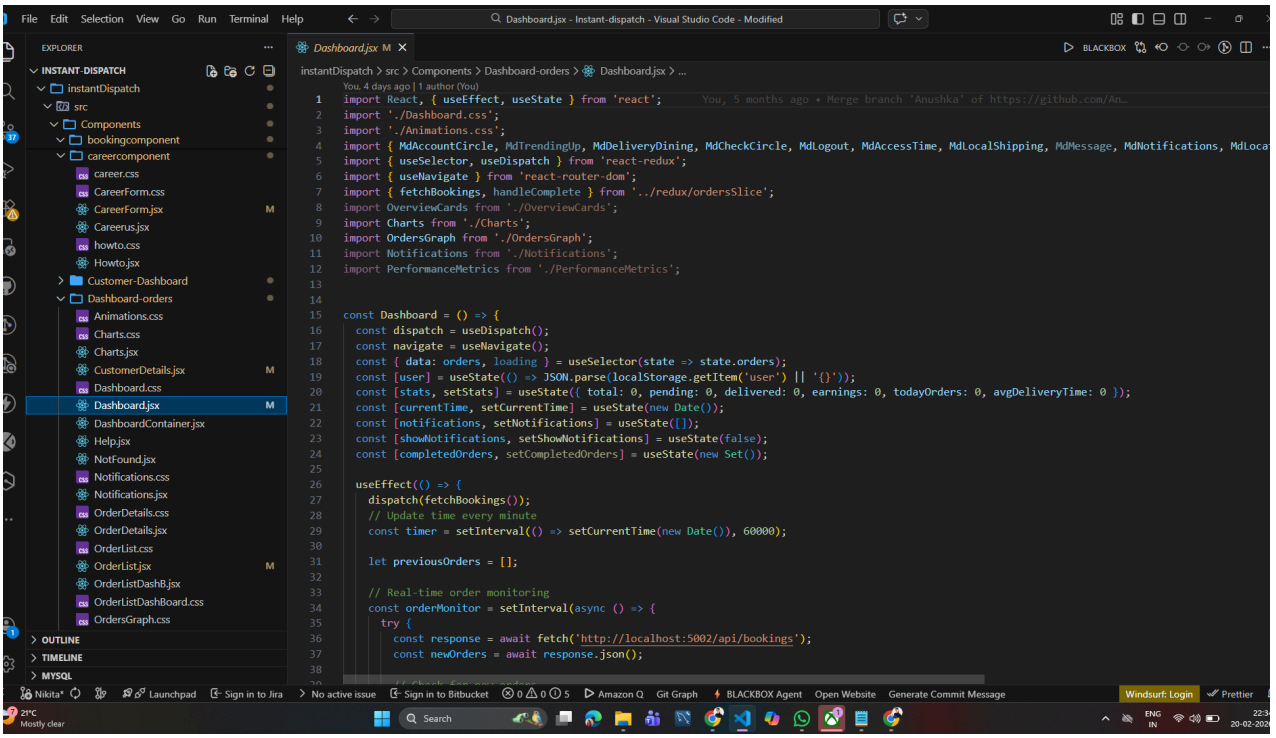


Fig. Rider Dashboard code

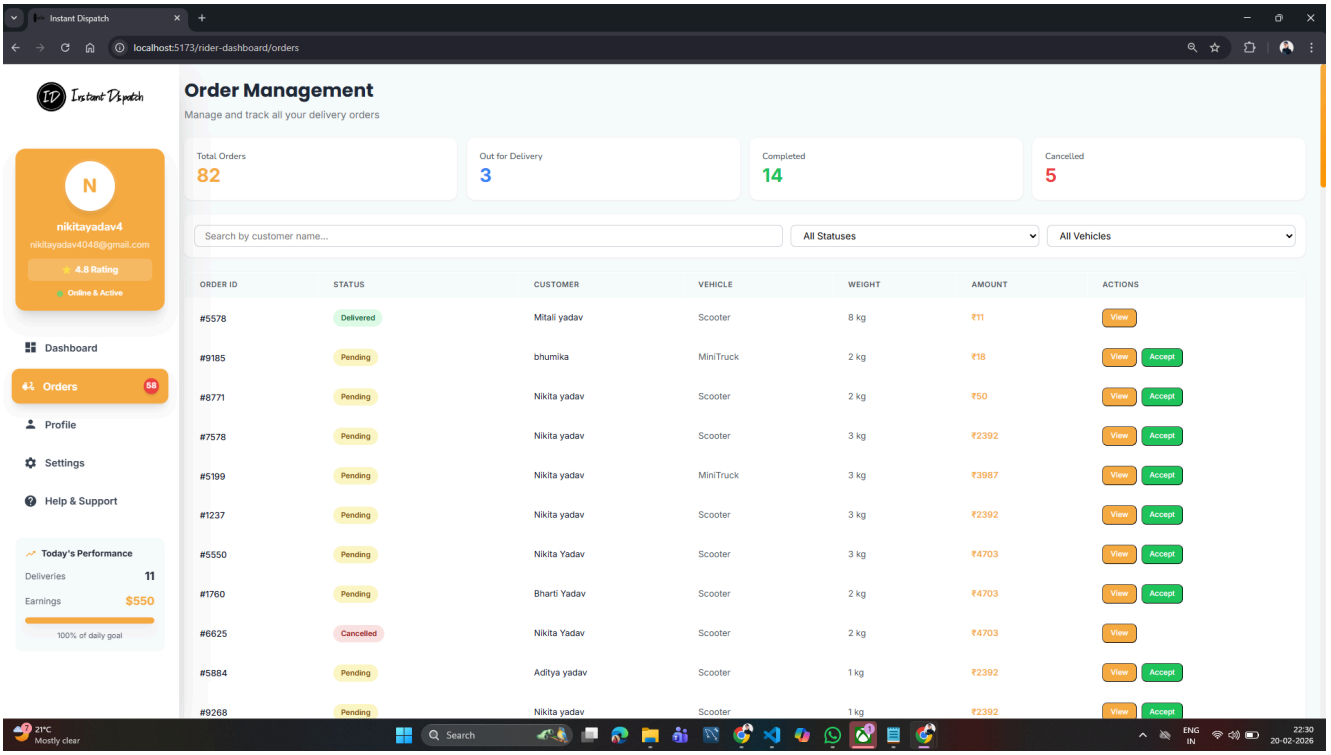


Fig. Order List

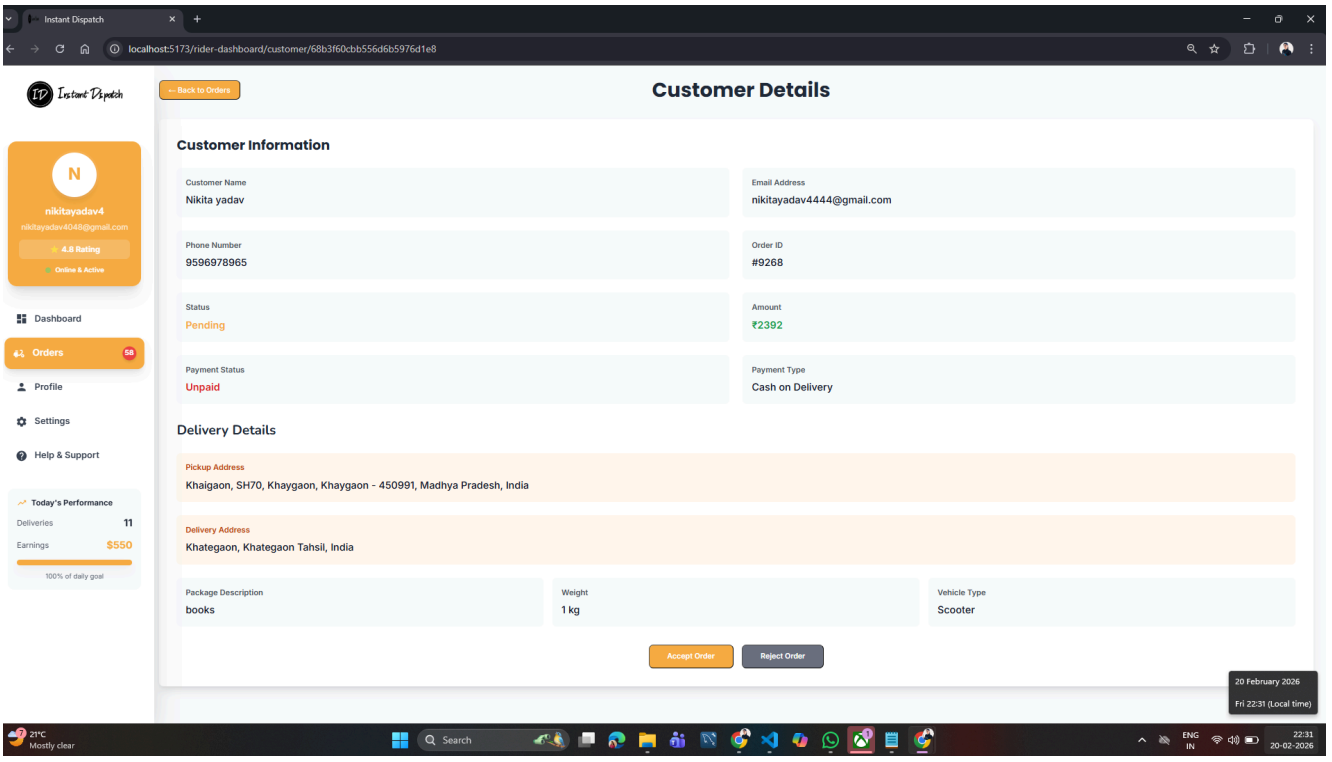


Fig. Order Details

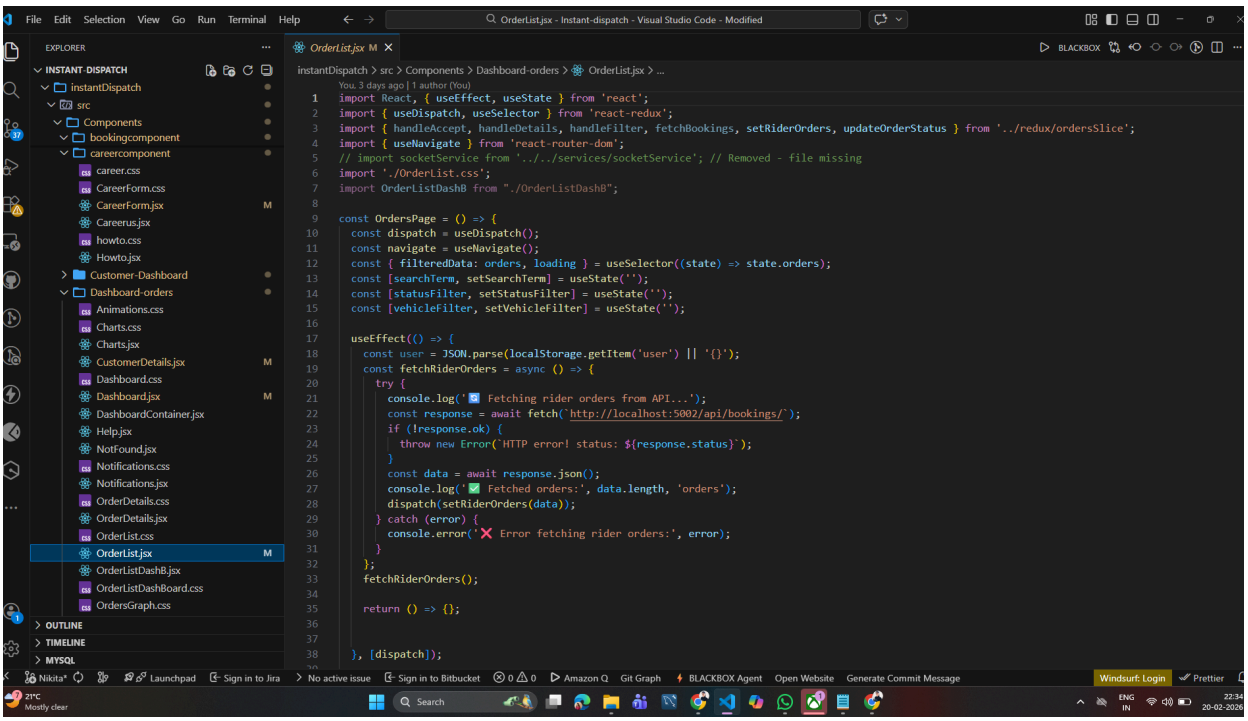


Fig. Order List Code

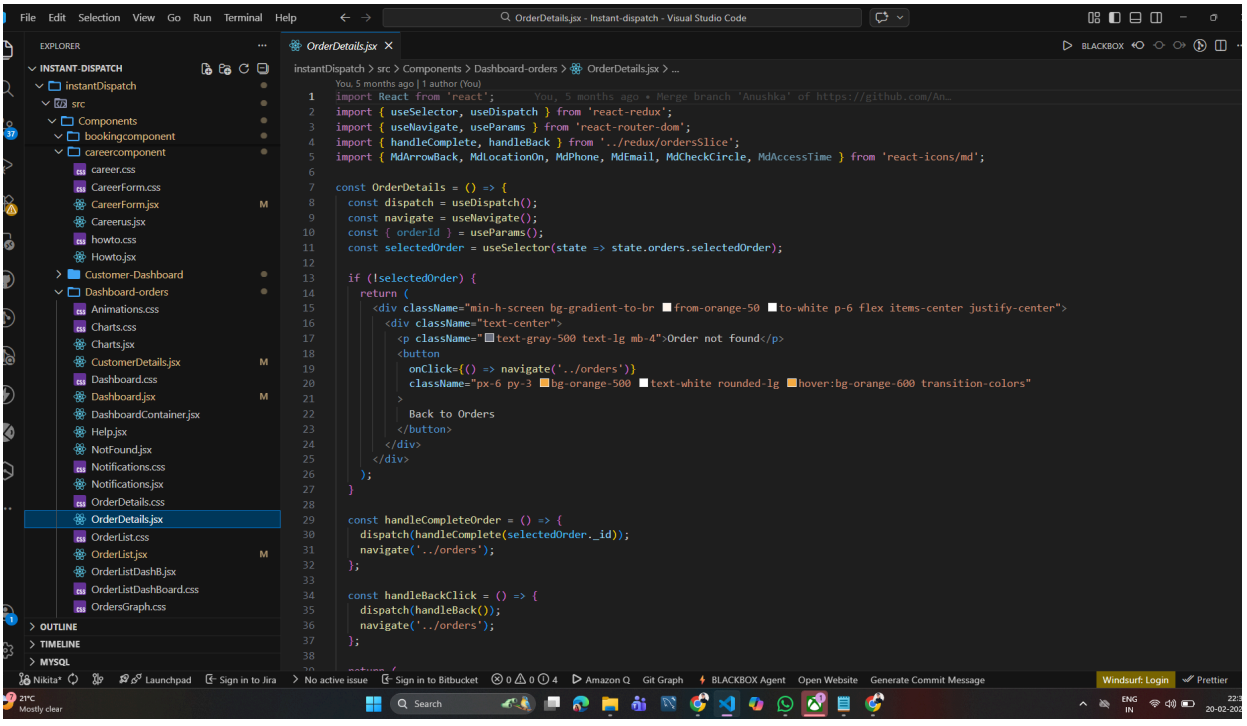


Fig.Order Details Code

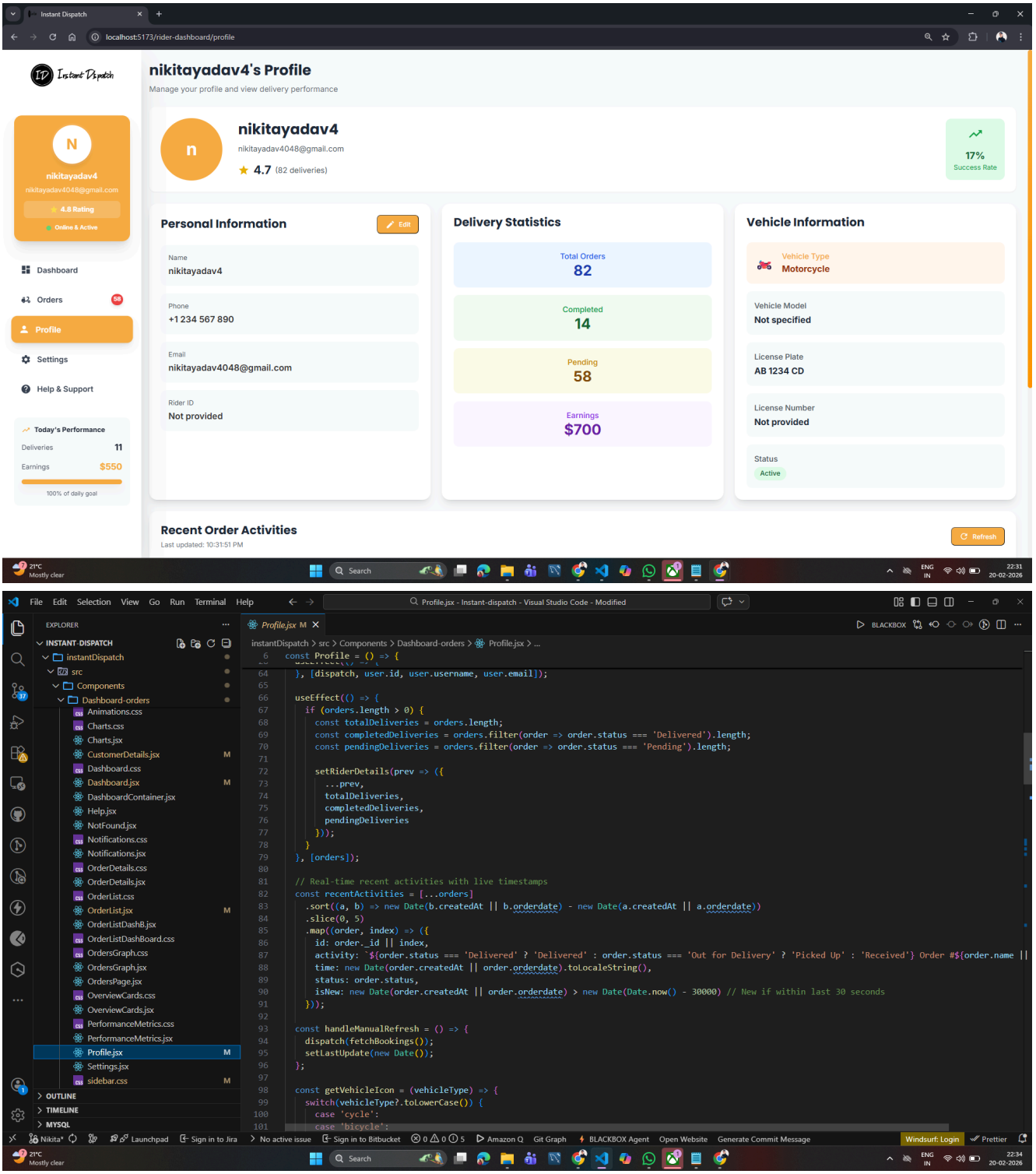
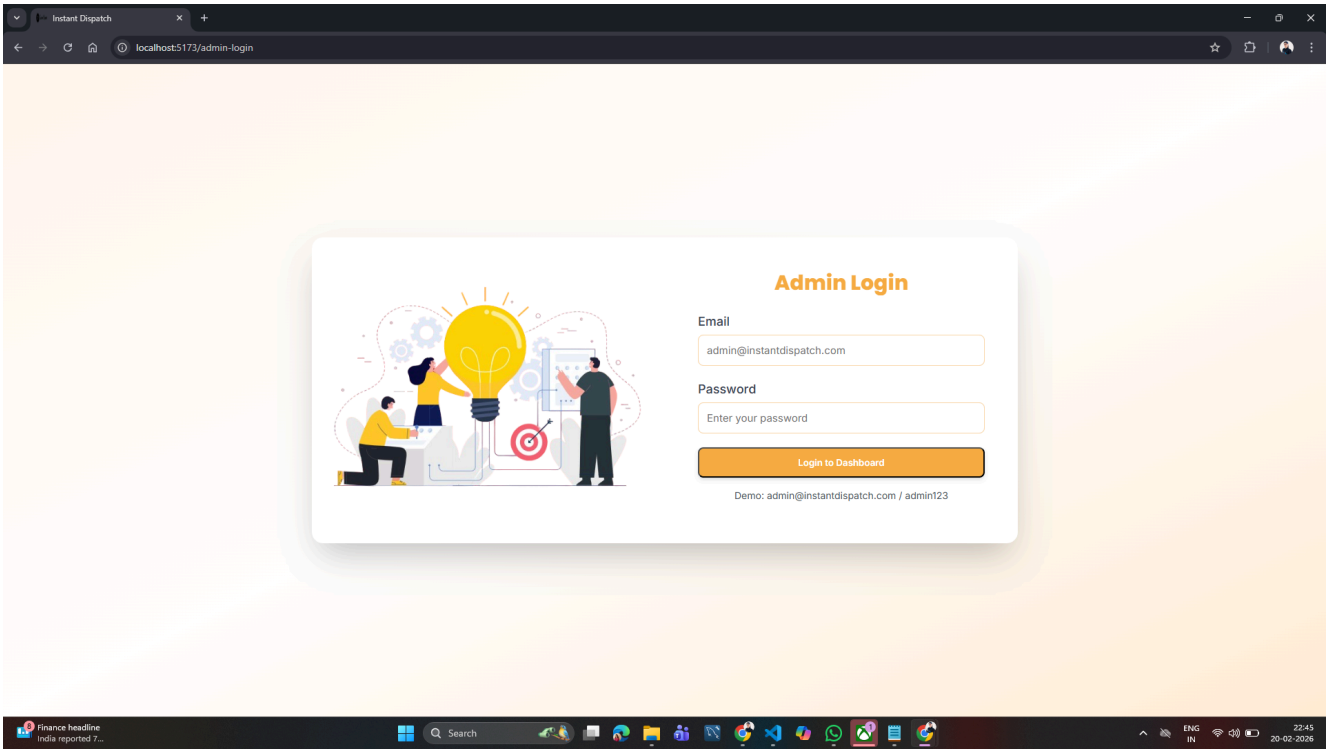


Fig. Rider Profile



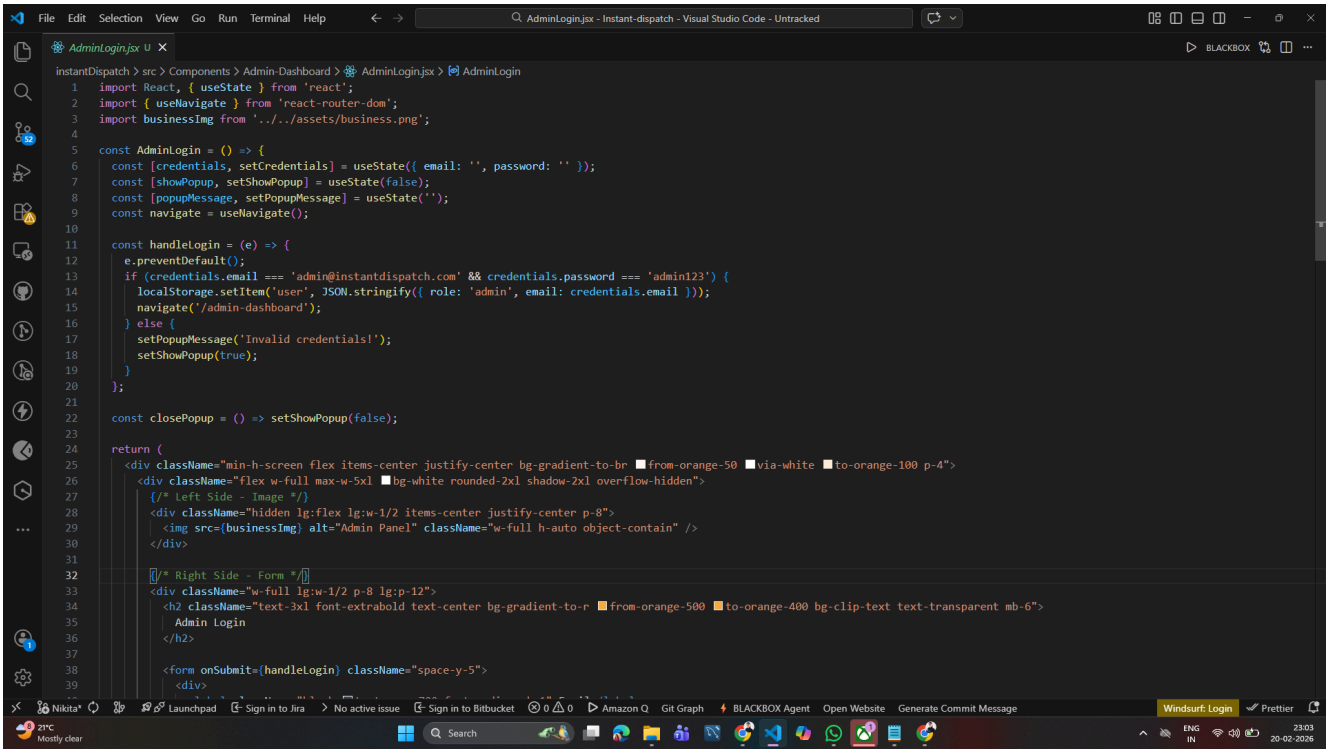
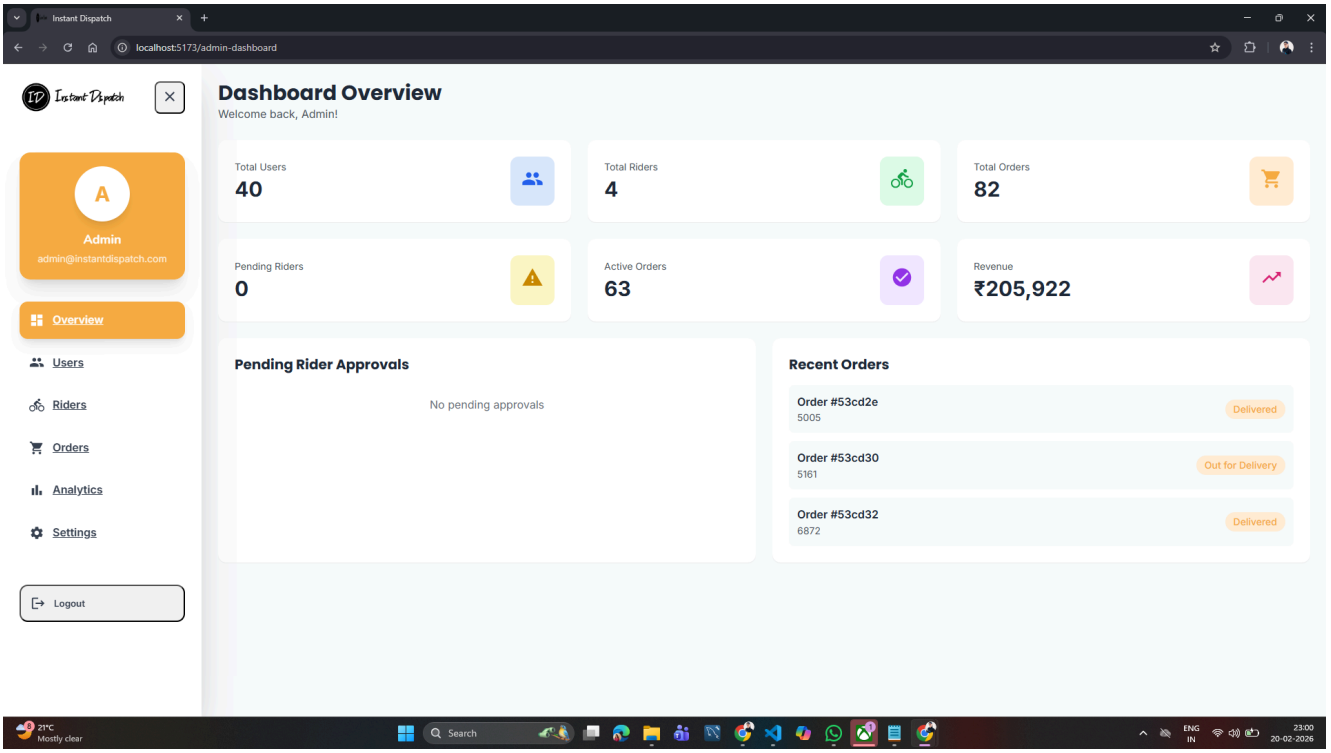


Fig. Rider Profile



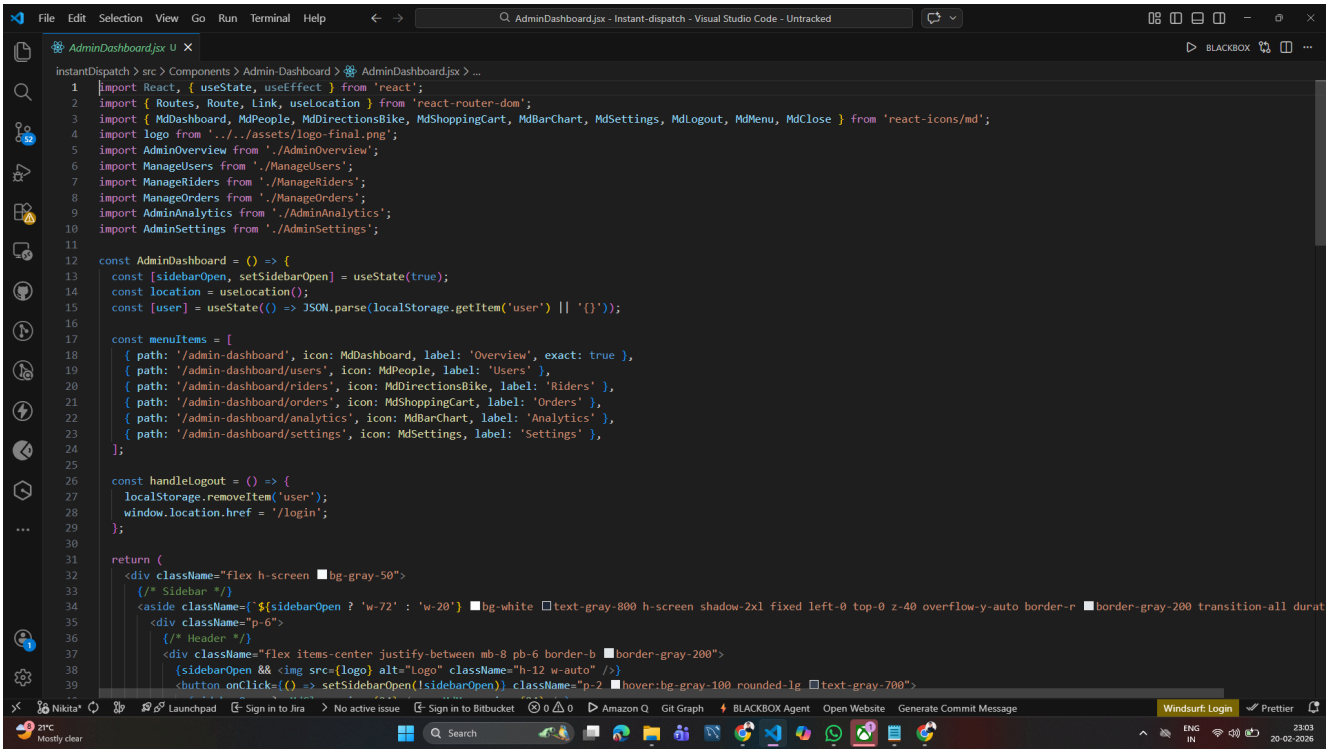
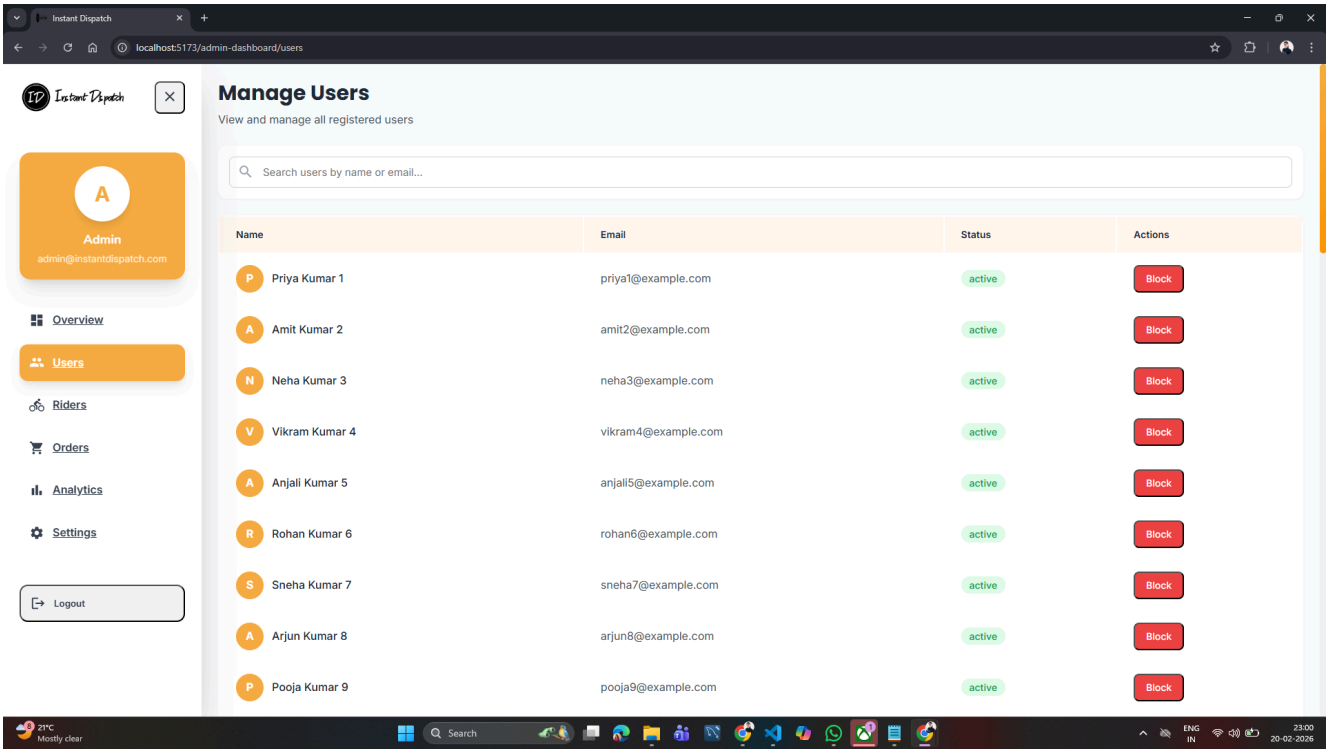


Fig.Admin Dashboard



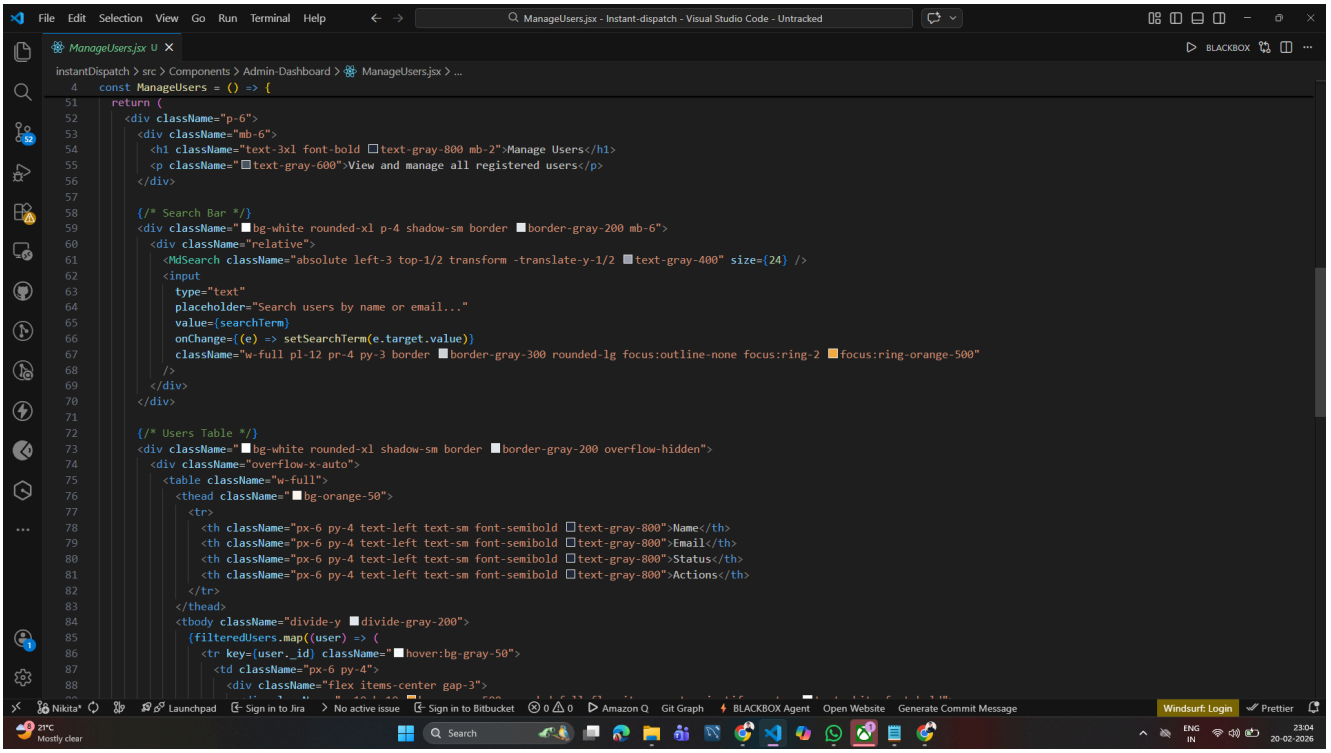
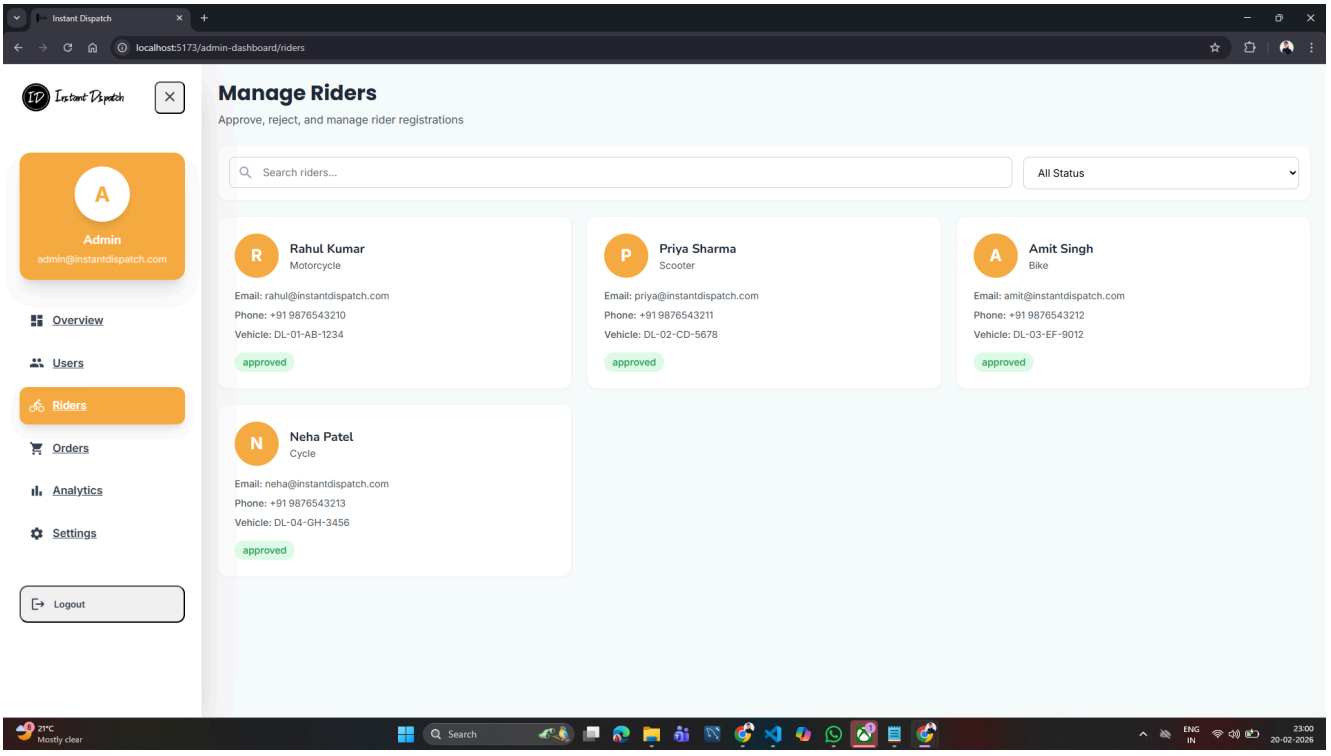


Fig. Admin User Management



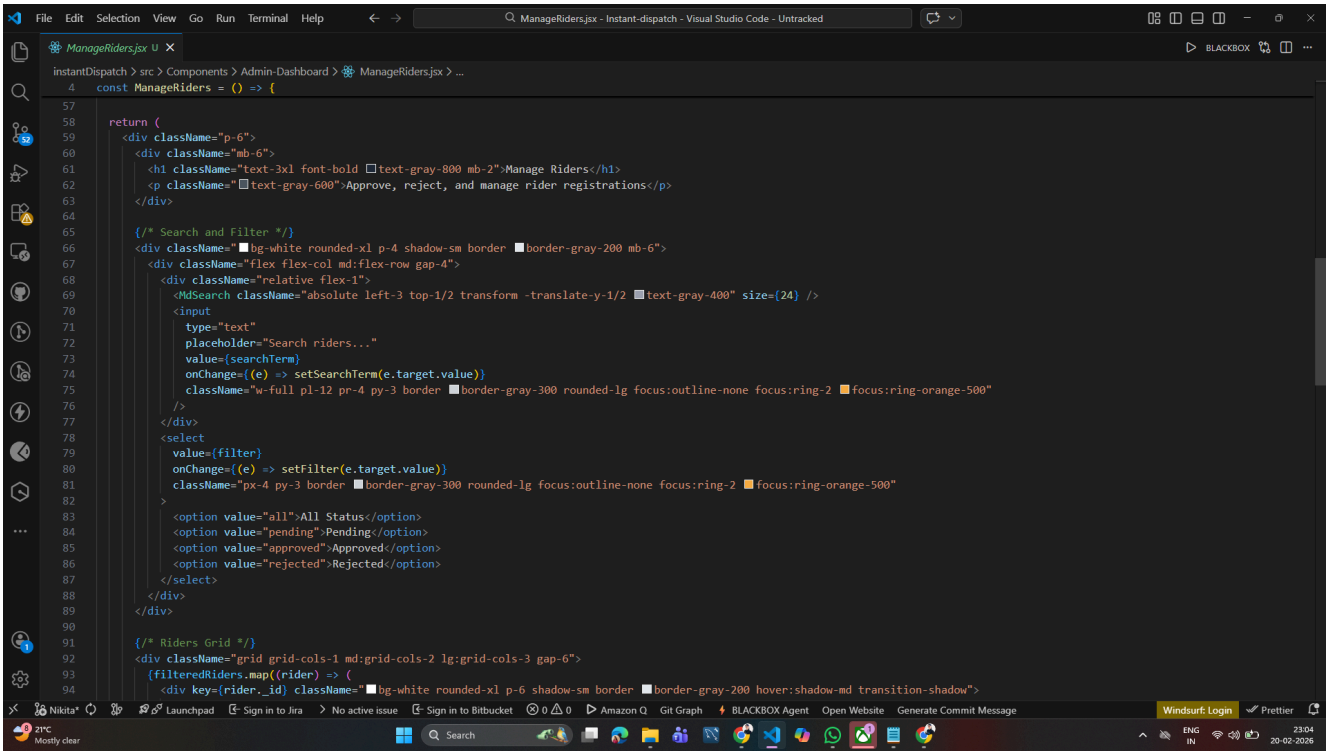
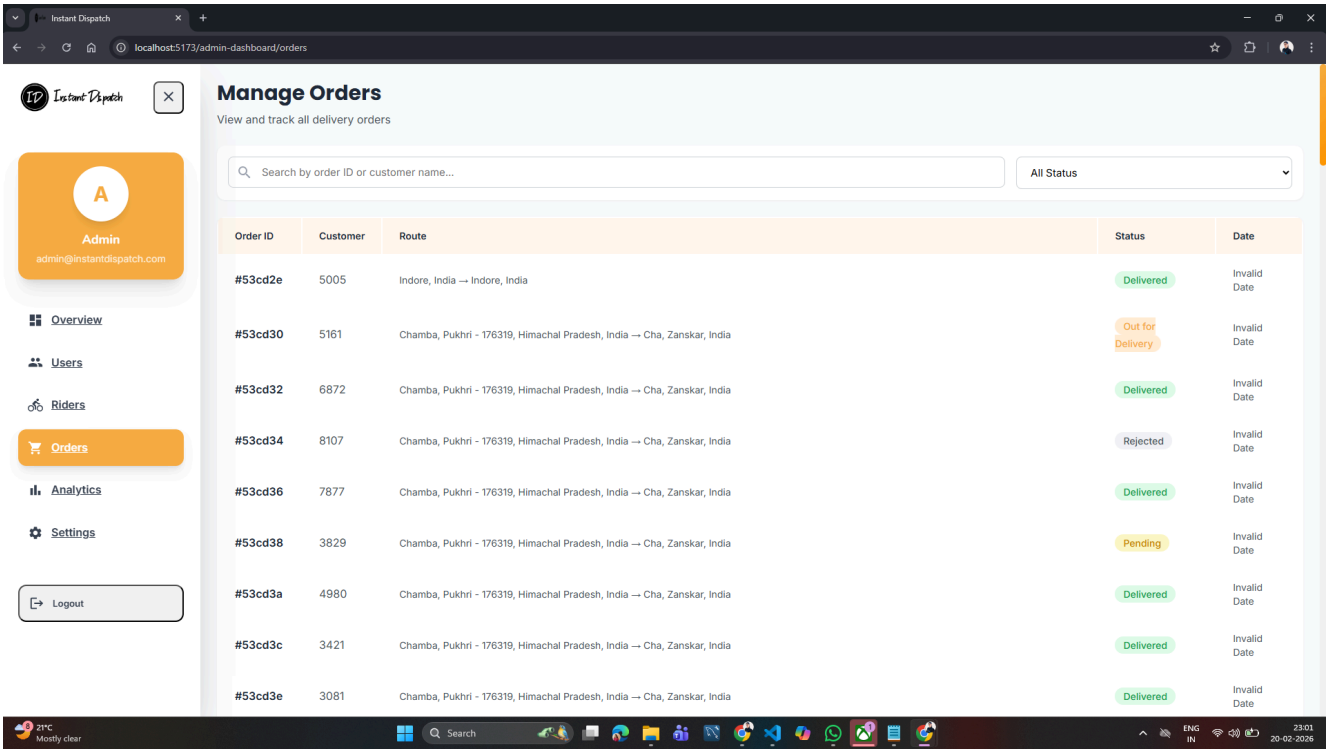


Fig. Admin Rider Management



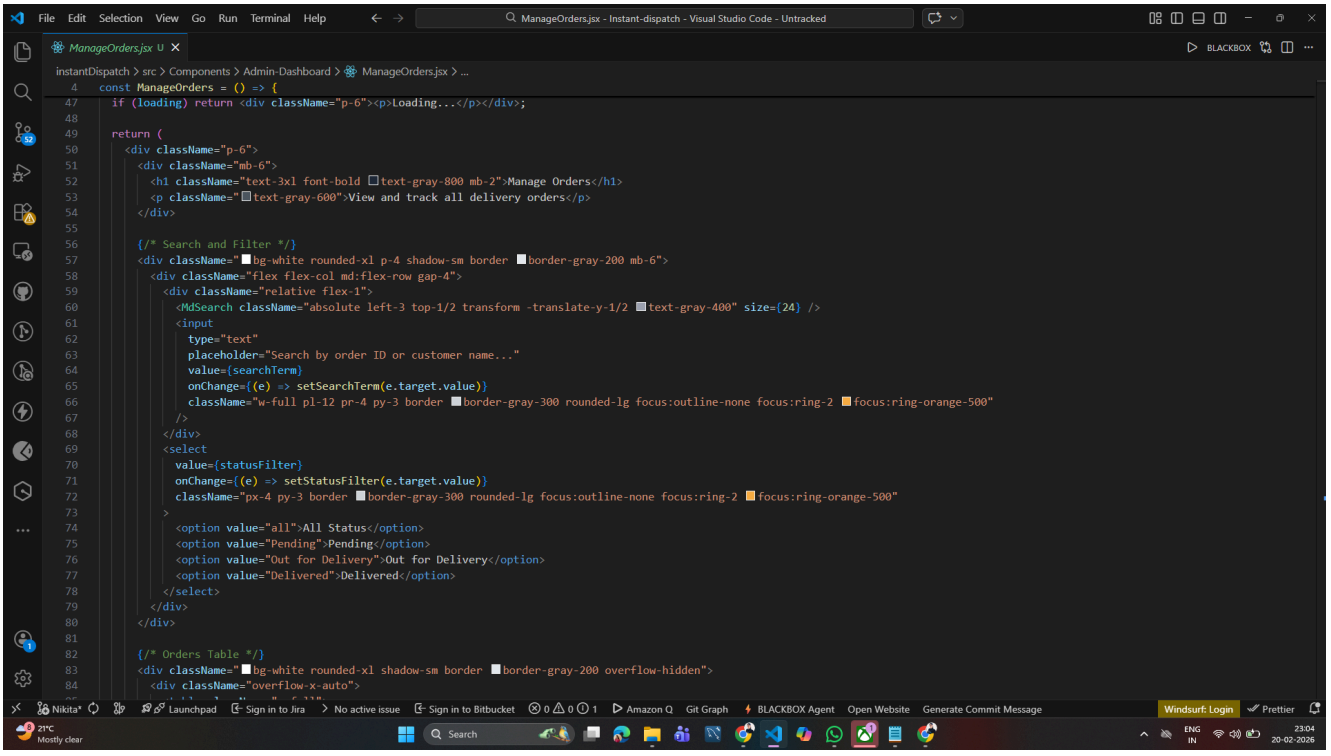


Fig. Admin Order Management

Conclusion

Project Summary

Instant Dispatch successfully provides a comprehensive platform for delivery management, connecting customers with reliable riders. The platform offers:

- User-friendly interface
- Real-time tracking
- Secure authentication
- Efficient order management
- Admin control panel

7.1 Limitations

In our "Instant Dispatch" we are still improving and trying to add new functionalities:

- Limited payment methods (currently Cash on Delivery and basic online payment)
- No multi-language support yet
- Limited to specific geographic regions
- Basic analytics dashboard
- No mobile application (web-only)

7.2 Future Enhancements

While Instant Dispatch is committed to continuous improvement, planned features include:

- Mobile Applications: Native iOS and Android apps
- Advanced Payment Options: Multiple payment gateways, wallet integration
- AI-based Route Optimization: Smart routing for faster deliveries
- Multi-language Support: Support for regional languages

- Advanced Analytics: Detailed reports and insights
- Rating System Enhancement: Detailed feedback mechanism
- Scheduled Deliveries: Book deliveries in advance
- Bulk Booking: Multiple deliveries at once
- Loyalty Programs: Rewards for frequent users
- Live Chat Support: Real-time customer support
- Push Notifications: Mobile and web push notifications
- Document Verification: Automated rider verification
- Insurance Integration: Package insurance options

Bibliography

1. https://react.dev/learn/
2. https://nodejs.org/en/docs/
3. https://expressjs.com/
4. https://mongoosejs.com/docs/
5. https://socket.io/docs/
6. https://redux-toolkit.js.org/
7. https://stackoverflow.com/
8. https://github.com/
9. https://tailwindcss.com/docs
10. https://www.mongodb.com/docs/