e-Conductor

30th November 2024

PROJECT REPORT

Group 09

E/20/032 Bandara A.M.N.C.

E/20/035 Bandara K.C.H.N.A.W.M.R.C.J.N.

E/20/133 Haththella H.A.D.T.N.

E/20/157 Janakantha S.M.B.G.

Supervisor:

Prof. Manjula Sandirigama

Table of Content

Introduction

e-Conductor is an online bus ticketing platform for passengers, bus owners, and transport authorities. It includes a mobile app for ticket booking, schedules, and real-time updates, plus an admin web portal for bus tracking, schedule, and revenue management. The platform provides a seamless experience through the following components.

Mobile App: Available for passengers, bus owners, and operators. Passengers can book tickets, view schedules, and get real-time updates, while bus owners and operators can manage their buses, track operations, and monitor performance.

Website: Accessible to passengers, bus owners, and operators for ticket bookings, schedule management, and bus monitoring.

Admin Web Portal: Exclusively for transport authorities, offering advanced tools for bus tracking, schedule management, and revenue analysis.

Backend: A secure and scalable system supporting data management, communication, and operational integration.

Objectives

- Simplify ticket booking and scheduling for passengers and bus operators.
- Provide real-time updates to improve journey planning and management.
- Empower bus owners and operators with tools to streamline operations and track performance.
- Enable transport authorities to manage and analyse data for better decision-making.

Functional Requirements

The e-Conductor system supports various user roles (passengers, bus owners, operators, and transport authorities) with the following key features and functionalities:

For Passengers

- Ticket Booking: Book single or multiple tickets via the mobile app or website
- Ticket Cancellation: Cancel booked tickets and get a refund
- Schedule Access: View available buses, routes, and seat availability
- Real-Time Updates: Receive live bus locations
- Payment Integration: Pay securely through google pay
- Ticket Management: View and manage booked tickets

For Bus Owners

- Revenue Monitoring: Track individual bus earnings and analytics.
- Schedule Management: Update, cancel or replace schedules
- Bus Management: Register new buses, update bus details and manage bus operators
- Real-Time Updates: Receive live bus locations of owned buses

For Bus Operators

- Booked Ticket Management: View all booked seats for their buses
- Ticket Validation: Scan ticket QR codes using the mobile app to verify ticket authenticity.
- Location Sharing: Continuously send the bus's real-time location to the backend system for passenger updates and tracking.

For Transport Authorities (Admin web portal)

- Schedule Oversight: Approve and manage schedules submitted.
- Revenue Monitoring: Track overall system revenues and analytics.
- Compliance Monitoring: Ensure users of the system adhere to regulations.
- System Management: Manage user roles, permissions, and overall platform integrity.

Non-Functional Requirements

Security

Requirement: Ensure secure user authentication, data protection, and payment transactions.

Implementation:

- Deployed OTP-based Authentication for secure login and verification.
- Used Google Authentication (OAuth 2.0) for secure alternative logins and account management.
- Implemented data encryption using HTTPS for secure communication and secure storage for sensitive information like passwords and payment details.
- Role-based access control to restrict unauthorized access to sensitive features and data.

Performance

Requirement: Deliver a responsive, low-latency system capable of handling high user demand.

Implementation:

- Configurable location tracking intervals to reduce frequent updates and optimize battery usage.
- Backend data caching: Certain data is stored locally in files on the client side to minimize redundant API calls.
- Transitioned the backend from Google Distance Matrix API to the Directions API, improving both performance and data accuracy.

Usability

Requirement: Ensure an intuitive and accessible platform for diverse users.

Implementation:

- User-friendly and responsive UI/UX design tested for ease of use.
- Consistent performance across mobile and web platforms, ensuring seamless user experiences.

Payment Integration

Requirement: Enable reliable and secure payment processing.

Implementation:

- Integrated Google Pay for seamless, contactless payments.
- Used Stripe for PCI DSS-compliant payment processing in the mobile app.

- Built-in real-time error handling and confirmation mechanisms for payment operations.

Scalability

Requirement: Adapt to growing user demands and evolving system requirements.

Implementation:

- Modular architecture for seamless addition of new features.

Use Case Descriptions

Booking a Bus Ticket (Mobile app)

Primary Actor: Passenger

Goal: Book a bus ticket for a selected route and schedule.

Preconditions: The passenger must be logged into the mobile app.

Main Flow:

1. The passenger selects from, to locations and date from available options.
2. The system displays available bus details and ticket prices.
   - If the departure time - current time is less than 24 hours, the system allows the passenger to book a Quick Ticket, and seat selection is not available.
   - If the departure time is more than 24 hours away, the system allows the passenger to select a seat and book a Normal Ticket.
3. Quick Ticket Flow (if within 24 hours):
   - The passenger chooses a bus
   - The passenger chooses full and half ticket count and proceeds to payment.
   - The system processes the payment using system credits and issues a ticket.

4. Normal Ticket Flow (if more than 24 hours away):
   - The passenger chooses a bus, and the system displays its seat layout
   - The passenger chooses a seat, full and half ticket count and proceeds to payment.
   - The system processes the payment using system credits and issues a ticket.

Alternative Flows:

1. If the payment fails, the passenger is prompted to try again.

Postconditions: The passenger receives a ticket, and the system updates seat availability.

Scanning and Validating a Ticket (Mobile app)

Primary Actor: Bus Operator

Goal: Verify the validity of a passenger’s ticket during boarding.

Preconditions: The bus operator must be logged into the mobile app.

Main Flow:

1. The bus operator opens the ticket scanner in the app.
2. The operator scans the passenger’s ticket barcode or QR code.
3. The system checks the ticket against the booked seat data in the bus operators account to verify if it is valid.
4. The system displays the ticket’s status (valid or invalid) and the ticket details.

Alternative Flows:

1. If the ticket scan is failed, the system shows an error message

Postconditions: The ticket status is updated in the system, and the bus operator can allow or deny boarding.

Tracking a Bus (Web App)

Primary Actor: Passenger

Goal: Track the real-time location of a bus for a booked journey.

Preconditions:

- The passenger must be logged into the web app.
- The passenger must have an active booking for the bus journey.

Main Flow:

1. The passenger navigates to the "Available Tickets" section of the web app.
2. The system displays the passenger's active bookings with options to track the bus.
3. The passenger selects the "Track Bus" option for a specific booking.
4. The system fetches the real-time location of the bus based on its GPS data.
5. The system displays the bus's current location on a map with relevant details, such as estimated time of arrival (ETA) and distance from the passenger’s boarding point.

Alternative Flows:

- If the bus tracking data is unavailable due to technical issues or the bus being offline, the system displays a message: "Tracking data unavailable. Please try again later."
- If the passenger does not have an active booking, the system disables the "Track Bus" option and prompts: "No active bookings available to track."

Postconditions:

- The passenger views the bus’s real-time location and can make informed decisions based on the ETA.
- The system continues to update the bus's location periodically until the journey is complete.

Cancelling a Ticket (Web App)

Primary Actor: Passenger

Goal: Cancel a booked ticket and receive a refund based on the cancellation policy.

Preconditions:

- The passenger must be logged into the web app.

- The passenger must have an active ticket that is eligible for cancellation as per the cancellation policy.

Main Flow:

1. The passenger navigates to the "Available Ticket" section of the web app.
2. The system displays the passenger's active bookings with options to cancel tickets.
3. The passenger selects the "Cancel Ticket" option for a specific booking.
4. The system prompts the passenger to confirm the cancellation and displays the refund amount (if applicable) based on the cancellation policy.
5. The passenger confirms the cancellation.
6. The system processes the cancellation, updates the ticket status to "Cancelled," and initiates the refund process (if applicable).
7. The system notifies the passenger of the successful cancellation via an on-screen message and email.

Alternative Flows:

- If the ticket is ineligible for cancellation (e.g., past the cancellation window), the system displays a message: "This ticket cannot be cancelled as per the policy."
- If the cancellation process fails due to a system error, the system displays a message: "Cancellation failed. Please try again later."

Postconditions:

- The ticket is marked as "Cancelled," and the passenger receives a refund (if applicable).
- The seat associated with the cancelled ticket is made available for booking by other passengers.

Adding a Booking (Admin App)

Primary Actor: Admin

Goal: Manually adding a booking to the system in response to a customer request

Preconditions:

- The admin is logged into the admin portal.
- The bus schedule is already available in the system.
- Seats on the selected bus are available.

Main Flow:

1. Admin navigates to the **Booking Management** section of the admin app.
2. Admin clicks the **Add Booking** button.
3. Admin fills out the booking form, including Passenger details, Bus details and Seat.
4. Admin reviews the input data to ensure accuracy.
5. Admin clicks **Save** to confirm the booking.
6. System validates the data and updates the bus schedule with the reserved seat.
7. System generates a unique booking ID and saves the booking in the database.
8. Customer receives booking confirmation via email.

Postconditions:

1. The booking is successfully added to the system.
2. The customer receives confirmation of the booking.

Alternative Flows:

- **Condition:** Selected seats are already booked.
- **Steps:**
  - System displays an error message indicating the seat is unavailable.
  - Admin selects different available seats and continues the booking process.

- **Condition:** Admin does not provide mandatory passenger details (e.g., name, contact info).
- **Steps:**
  - System displays a validation error.
  - Admin provides the required details and retries.

Test Cases

Ticket Scanning for Validity (Bus Operator)

Test Objective: Verify that the bus operator can scan a passenger’s ticket and validate its authenticity (whether it's a valid ticket for the journey).

Test Steps:

1. Book a ticket using a passenger account
2. Log in to the account of the bus operator of the bus that ticket belongs to using a different device
3. The bus operator scans the QR code or barcode on the passenger’s ticket.
4. The system checks the bus registration number, departure time, seat number
5. The system displays ticket status (valid or invalid) and other details

Expected Outcome:

- For valid tickets: The system confirms the validity of the ticket, allowing the passenger to board.
- For invalid tickets: The system alerts the operator with an error message and prevents the passenger from boarding.

Bus Tracking Feature Verification (Passenger)

Test Objective:

Verify that the bus tracking feature accurately displays the current location of the bus on the passenger's interface.

Test Steps:

1. Log in as a passenger and book a ticket for a specific bus.
2. Navigate to the **Available Ticket** section in the menu.
3. Confirm that the bus registration number and departure details match the booked ticket.
4. Start the journey simulation for the bus on the backend or live system.
5. Monitor the displayed bus location on the map interface in real-time.
6. Simulate changes in the bus location dynamically.

Expected Outcome:

- The map should accurately display the bus's current location and update in real-time.
- If the bus reaches the destination or the distance becomes negative, the tracking should stop with an appropriate message: "Journey completed" or "Tracking stopped."
- In case of any errors in fetching the location, the system should display an error message, e.g., "Unable to fetch live location."

Ticket Cancellation Process Verification (Passenger)

Test Objective:

Verify that the ticket cancellation process functions correctly, including refund processing and seat availability updates.

Test Steps:

1. Log in as a passenger and book a ticket for a specific journey.
2. Navigate to the **Available Tickets** section in the menu.
3. Select a booked ticket and initiate the cancellation process.
4. Confirm the cancellation when prompted by the system.
5. Check the updated seat availability for the corresponding journey in the system.
6. Verify the refund status in the passenger's account or linked payment method.

Expected Outcome:

- The ticket is successfully cancelled, and a confirmation message, e.g., "Ticket cancelled successfully," is displayed.
- The cancelled seat becomes available for new bookings.
- The refund status is updated correctly, with a message indicating the refund amount and processing timeline.
- In case of errors, the system provides an appropriate error message, e.g., "Unable to process cancellation."

OTP Validity (Backend)

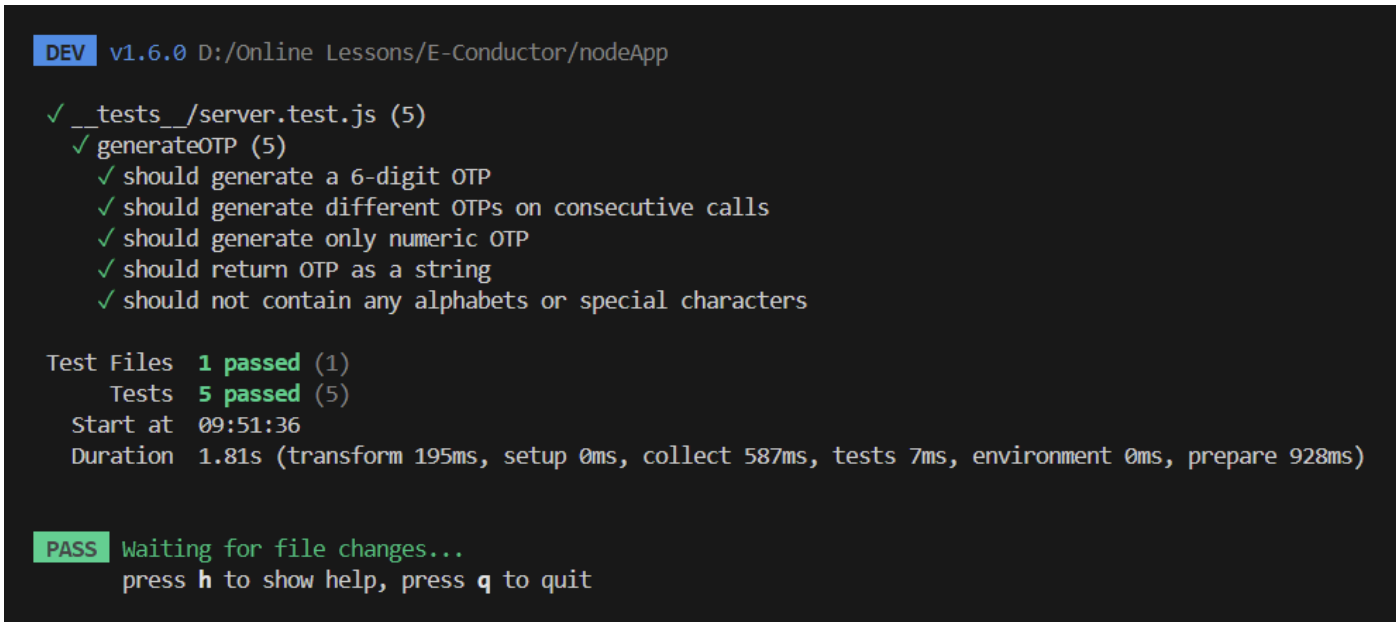
Test Objective:

Verify that the OTP is generator function output is in the required format.

Test Steps:

1. Check that the OTP is 6 digits long.
2. Check that OTP consists only of numbers.
3. Ensure consecutive OTPs are not the same.
4. Ensure it is a string and contains only numbers.
5. Ensure no alphabets or special characters are present.

Expected Outcome:



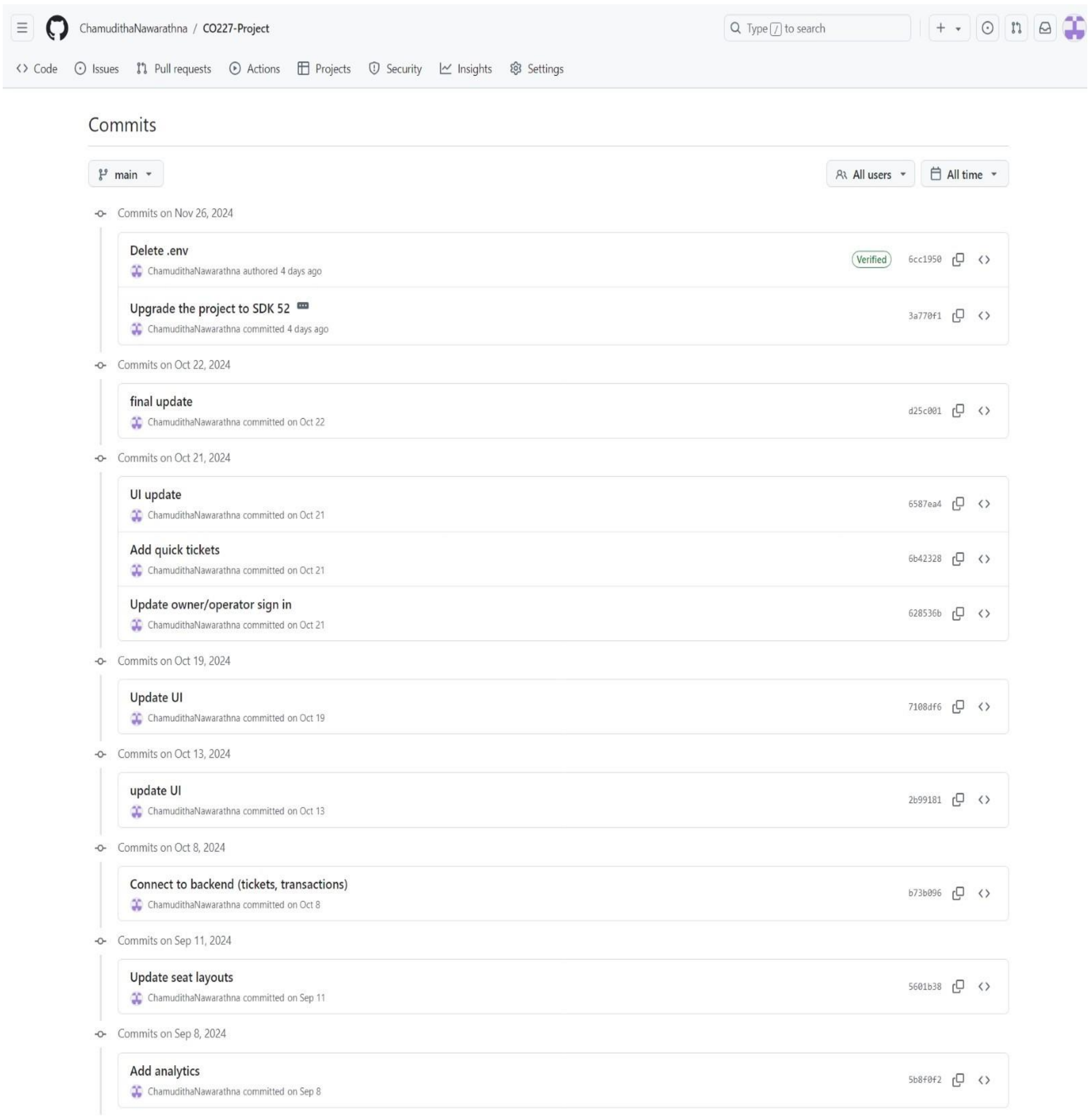
```
 DEV  v1.6.0 D:/Online Lessons/E-Conductor/nodeApp

 ✓ __tests__/server.test.js (5)
   ✓ generateOTP (5)
     ✓ should generate a 6-digit OTP
     ✓ should generate different OTPs on consecutive calls
     ✓ should generate only numeric OTP
     ✓ should return OTP as a string
     ✓ should not contain any alphabets or special characters

 Test Files  1 passed (1)
      Tests  5 passed (5)
   Start at  09:51:36
   Duration  1.81s (transform 195ms, setup 0ms, collect 587ms, tests 7ms, environment 0ms, prepare 928ms)


 PASS  Waiting for file changes...
       press h to show help, press q to quit
```

Project Management and Version Control

Mobile app

Project link: ChamudithaNawarathna/CO227-Project
Project link (main): e-Conductor_V2.0/05. Mobile App at main · Bimsara-Janakantha/e-Conductor_V2.0

Screenshot of commit history:



ChamudithaNawarathna / CO227-Project

Commits

main · All users · All time

Commits on Nov 26, 2024

Delete .env
ChamudithaNawarathna authored 4 days ago — Verified — 6cc1950

Upgrade the project to SDK 52
ChamudithaNawarathna committed 4 days ago — 3a770f1

Commits on Oct 22, 2024

final update
ChamudithaNawarathna committed on Oct 22 — d25c001

Commits on Oct 21, 2024

UI update
ChamudithaNawarathna committed on Oct 21 — 6587ea4

Add quick tickets
ChamudithaNawarathna committed on Oct 21 — 6b42328

Update owner/operator sign in
ChamudithaNawarathna committed on Oct 21 — 628536b

Commits on Oct 19, 2024

Update UI
ChamudithaNawarathna committed on Oct 19 — 7108df6

Commits on Oct 13, 2024

update UI
ChamudithaNawarathna committed on Oct 13 — 2b99181

Commits on Oct 8, 2024

Connect to backend (tickets, transactions)
ChamudithaNawarathna committed on Oct 8 — b73b096

Commits on Sep 11, 2024

Update seat layouts
ChamudithaNawarathna committed on Sep 11 — 5601b38

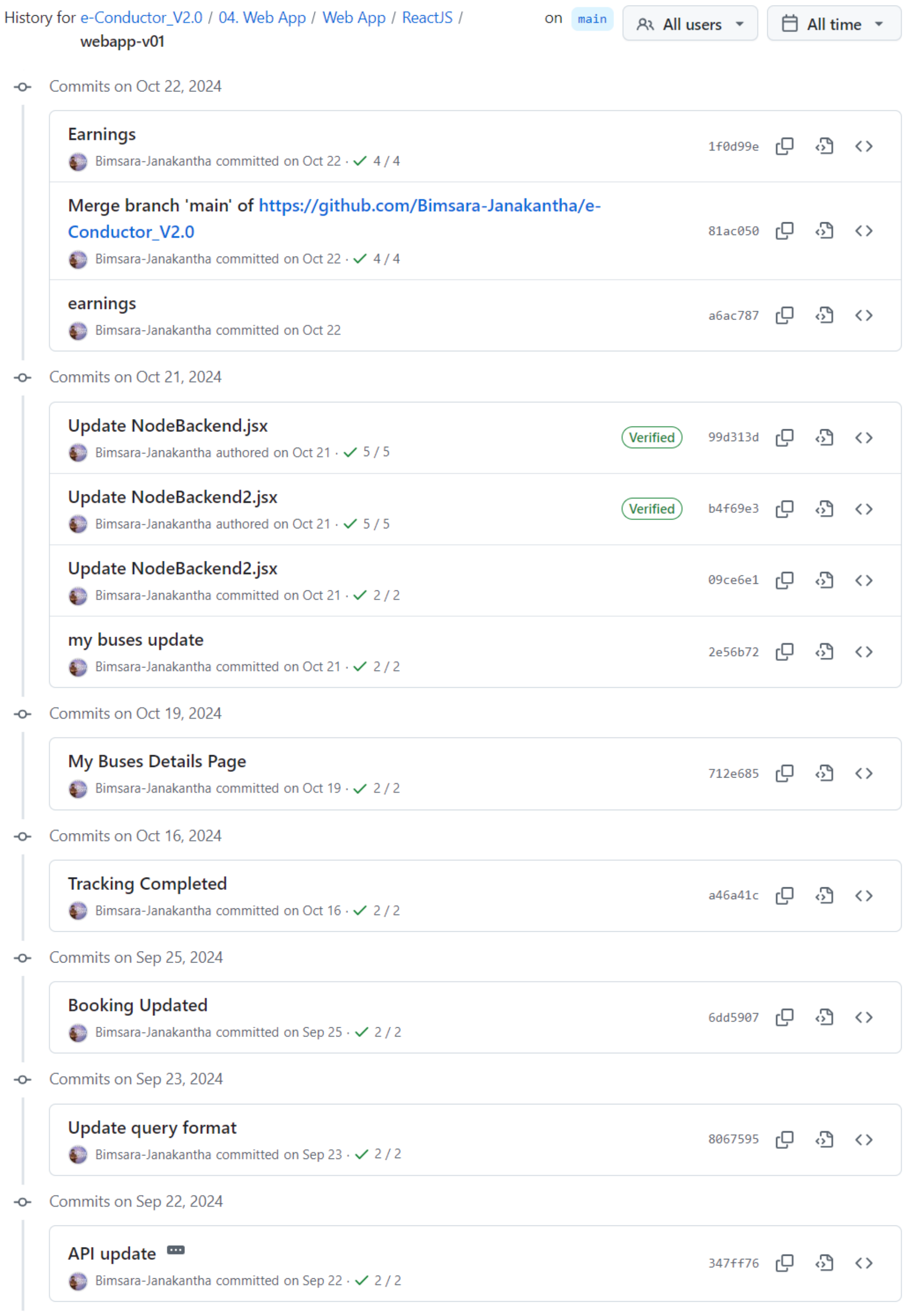
Commits on Sep 8, 2024

Add analytics
ChamudithaNawarathna committed on Sep 8 — 5b8f0f2

Website

Deployment link: https://e-conductor-v2-0.pages.dev/home
Project Link: https://github.com/Bimsara-Janakantha/e-Conductor_V2.0/tree/main/04.%20Web%20App/Web%20App/ReactJS/webapp-v01

Screenshot of commit history:



History for e-Conductor_V2.0 / 04. Web App / Web App / ReactJS / webapp-v01 on main

Commits on Oct 22, 2024

Earnings
Bimsara-Janakantha committed on Oct 22 · ✓ 4 / 4 — 1f0d99e

Merge branch 'main' of https://github.com/Bimsara-Janakantha/e-Conductor_V2.0
Bimsara-Janakantha committed on Oct 22 · ✓ 4 / 4 — 81ac050

earnings
Bimsara-Janakantha committed on Oct 22 — a6ac787

Commits on Oct 21, 2024

Update NodeBackend.jsx
Bimsara-Janakantha authored on Oct 21 · ✓ 5 / 5 — Verified — 99d313d

Update NodeBackend2.jsx
Bimsara-Janakantha authored on Oct 21 · ✓ 5 / 5 — Verified — b4f69e3

Update NodeBackend2.jsx
Bimsara-Janakantha committed on Oct 21 · ✓ 2 / 2 — 09ce6e1

my buses update
Bimsara-Janakantha committed on Oct 21 · ✓ 2 / 2 — 2e56b72

Commits on Oct 19, 2024

My Buses Details Page
Bimsara-Janakantha committed on Oct 19 · ✓ 2 / 2 — 712e685

Commits on Oct 16, 2024

Tracking Completed
Bimsara-Janakantha committed on Oct 16 · ✓ 2 / 2 — a46a41c

Commits on Sep 25, 2024

Booking Updated
Bimsara-Janakantha committed on Sep 25 · ✓ 2 / 2 — 6dd5907

Commits on Sep 23, 2024

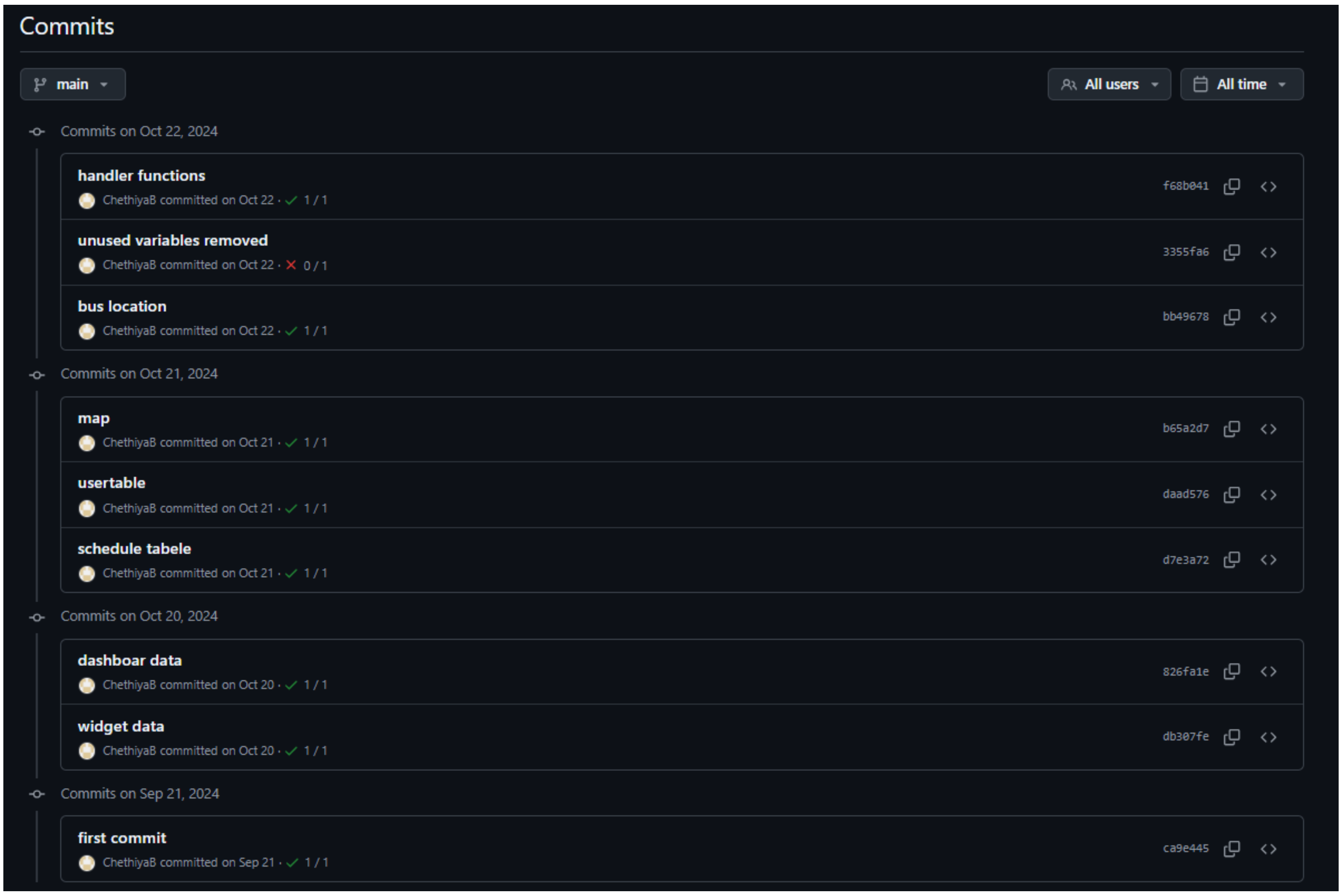
Update query format
Bimsara-Janakantha committed on Sep 23 · ✓ 2 / 2 — 8067595

Commits on Sep 22, 2024

API update
Bimsara-Janakantha committed on Sep 22 · ✓ 2 / 2 — 347ff76

Admin site

Project repository link: https://github.com/cepdnaclk/e20-co227-e-Conductor/tree/main/07.%20Admin%20App



Commits

main | All users | All time

Commits on Oct 22, 2024

handler functions
ChethiyaB committed on Oct 22 · ✓ 1 / 1 — f68b041

unused variables removed
ChethiyaB committed on Oct 22 · ✗ 0 / 1 — 3355fa6

bus location
ChethiyaB committed on Oct 22 · ✓ 1 / 1 — bb49678

Commits on Oct 21, 2024

map
ChethiyaB committed on Oct 21 · ✓ 1 / 1 — b65a2d7

usertable
ChethiyaB committed on Oct 21 · ✓ 1 / 1 — daad576

schedule tabele
ChethiyaB committed on Oct 21 · ✓ 1 / 1 — d7e3a72

Commits on Oct 20, 2024

dashboar data
ChethiyaB committed on Oct 20 · ✓ 1 / 1 — 826fa1e

widget data
ChethiyaB committed on Oct 20 · ✓ 1 / 1 — db307fe

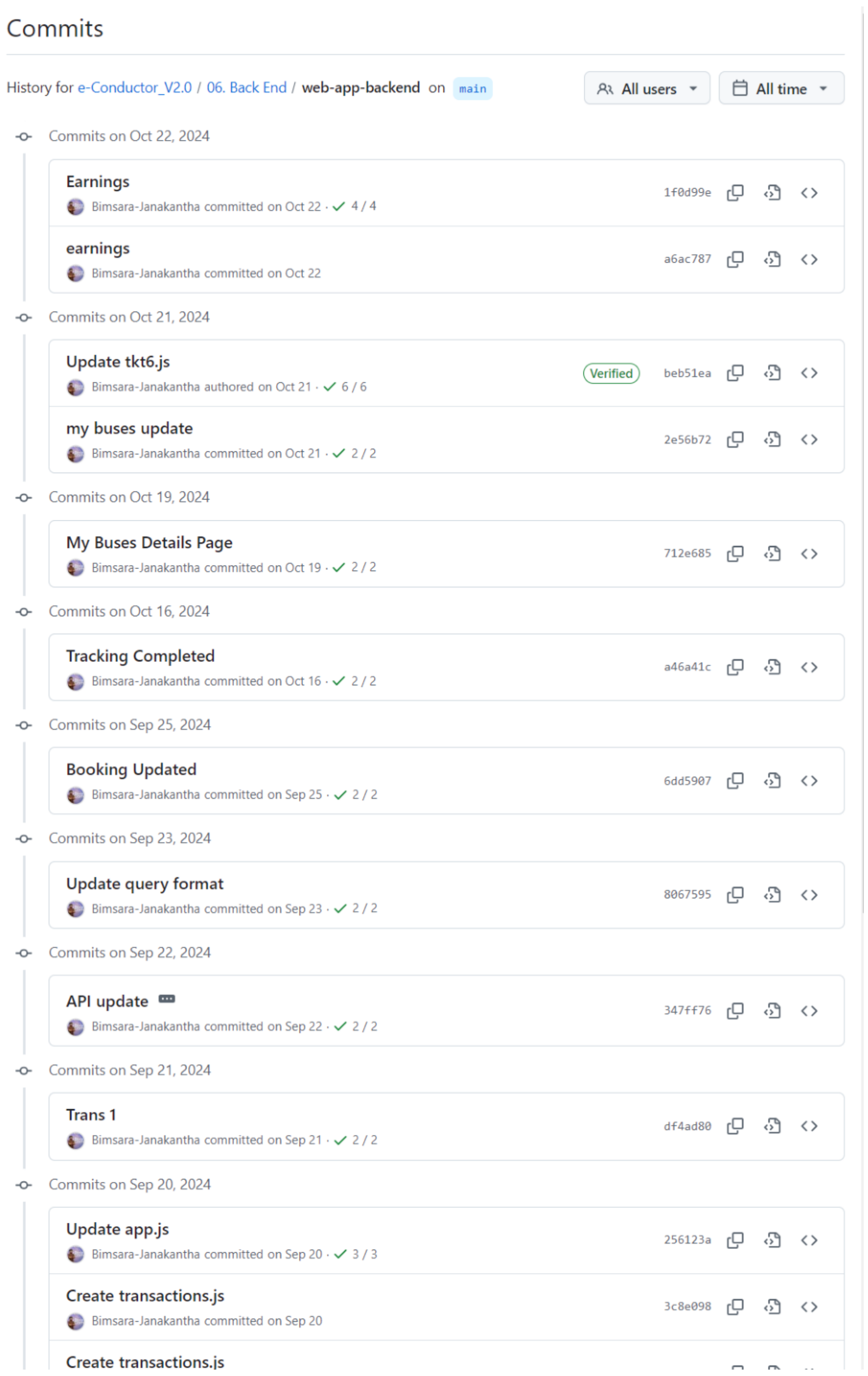
Commits on Sep 21, 2024

first commit
ChethiyaB committed on Sep 21 · ✓ 1 / 1 — ca9e445

Backend and database

Deployment link: https://webappbackend-production-e63c.up.railway.app
Project Link: https://github.com/Bimsara-Janakantha/e-Conductor_V2.0/tree/main/06.%20Back%20End/web-app-backend

Screenshot of commit history:


Commits

History for e-Conductor_V2.0 / 06. Back End / web-app-backend on main

All users · All time

Commits on Oct 22, 2024

Earnings
Bimsara-Janakantha committed on Oct 22 · ✓ 4 / 4 — 1f0d99e

earnings
Bimsara-Janakantha committed on Oct 22 — a6ac787

Commits on Oct 21, 2024

Update tkt6.js
Bimsara-Janakantha authored on Oct 21 · ✓ 6 / 6 — Verified — beb51ea

my buses update
Bimsara-Janakantha committed on Oct 21 · ✓ 2 / 2 — 2e56b72

Commits on Oct 19, 2024

My Buses Details Page
Bimsara-Janakantha committed on Oct 19 · ✓ 2 / 2 — 712e685

Commits on Oct 16, 2024

Tracking Completed
Bimsara-Janakantha committed on Oct 16 · ✓ 2 / 2 — a46a41c

Commits on Sep 25, 2024

Booking Updated
Bimsara-Janakantha committed on Sep 25 · ✓ 2 / 2 — 6dd5907

Commits on Sep 23, 2024

Update query format
Bimsara-Janakantha committed on Sep 23 · ✓ 2 / 2 — 8067595

Commits on Sep 22, 2024

API update
Bimsara-Janakantha committed on Sep 22 · ✓ 2 / 2 — 347ff76

Commits on Sep 21, 2024

Trans 1
Bimsara-Janakantha committed on Sep 21 · ✓ 2 / 2 — df4ad80

Commits on Sep 20, 2024

Update app.js
Bimsara-Janakantha committed on Sep 20 · ✓ 3 / 3 — 256123a

Create transactions.js
Bimsara-Janakantha committed on Sep 20 — 3c8e098

Create transactions.js

User Manual

Mobile app

Ticket Booking (Passenger)

1. Open the e-Conductor mobile app and log in to your passenger account.
2. Navigate to the **Tickets** tab on the **Home** screen
3. Enter the start, destination, and date of travel. Then click **Search**
4. The app displays available buses for your route. Showed buses would have **Buy Ticket** or **Quick Ticket.** If you want to travel within the next 24hrs, choose a bus with **Quick Ticket**. Otherwise, you can choose a bus with **Buy Ticket**.
5. If you chose **Buy Ticket,** you will be prompted to choose seats from the seat layout and enter how many full/half tickets you need. After completion, click **Book Seat**
6. If you chose **Quick Ticket**, you will be prompted to enter how many full/half tickets you need. After entering the number, click **Buy**.

   (Note: Quick Tickets option is provided for ticket purchasing after the booking period is over. This enables user to buy tickets while the bus is on route. However, there is no seat selection since after the booking period is over, we cannot guarantee a seat.)

Scanning Tickets (Bus Operator)

1. Open the e-Conductor mobile app and log in to your bus operator / employee account.
2. Navigate to the **Scanner** tab from the bottom tabs on the **Home** screen. When prompted, grant the app permission to use your device's camera.
3. Hold your mobile camera steady and point it at the ticket's QR code. Ensure the QR code is fully visible and well-lit for a clear view.
4. The scanner automatically retrieves the ticket details from the backend system. It verifies whether the ticket belongs to the bus you are currently assigned to, whether the ticket corresponds to today's schedule for this bus etc.
5. For valid tickets, a **Ticket Verified** message will appear, confirming the passenger's boarding. For invalid tickets, an error message will be displayed.

Tracking Bus Location (Bus Operator)

1. Open the e-Conductor mobile app and log in to your passenger account.
2. Navigate to the **Dashboard** tab on the **Home** screen
3. In the **Dashboard**, choose the desired location tracking time interval from the dropdown menu. The default interval is 2 seconds

   (Note: The smaller the time interval, the faster the battery will drain)

4. After setting the interval, press the **Start Tracking** button
5. To stop tracking, press the **Stop Tracking** button (which was previously the **Start Tracking** button).

Website

Accessing the System

1. Open a web browser and navigate to the system URL (https://e-conductor-v2-0.pages.dev).
2. Log in using your registered email and password.
3. If you don't have an account, click on Sign Up to create one.

Searching for Available Buses

1. On the **booking** page, enter the following details in the Search section:
   - Origin: Select the departure city.
   - Destination: Select the arrival city.
   - Travel Date: Choose the desired date of travel.
2. Click on **Continue**.
3. A list of available buses matching your criteria will be displayed.

Viewing Bus Details

1. Click on a bus from the search results to view its details:
   - Departure Time and Arrival Time.
   - Bus Type (e.g., AC, Non-AC).
   - Available Seats.
2. Click on **Continue** to proceed to seat selection.

Seat Selection

1. The Seat Selection interface displays the layout of available seats on the bus.
   - Available Seats are shown in *gray*.
   - Booked Seats are shown in *red*.
   - Driver Seat is marked with a *blue icon*.
2. Click on an Available Seat to select it.
   - The seat will turn *green*, indicating it is selected.
3. If you need to change your selection:
   - Click the selected seat again to deselect it.
   - Choose a different seat.

Booking Confirmation

1. Once seats are selected, click **Continue**.
2. Review your booking details, including:
   - Selected seats.
   - Total price.
   - Bus and journey details.
   - etc.
3. Choose a payment method and complete the payment.

Viewing Booked Tickets

1. Navigate to the **Available Tickets** section from the menu.
2. Click on a booking to:
   - View the ticket details.
   - Download or print the ticket.

Canceling a Booking

1. Go to **Available Tickets**.
2. Select the booking you wish to cancel.
3. Click **Cancel** button and confirm the cancellation.

Admin site

User Management

1. Log in to the admin portal.
2. Navigate to the User Management section.
3. You will see a list of users, including passengers and bus operators.
4. For each user, you can:
   - **View Profile**: Access user details such as name, email, and phone number.
   - **Update Information**: Edit user details if necessary.
   - **Terminate Account**: Temporarily block access to a user account.

Manage Bookings

1. Navigate to the Booking Management section.
2. View all ticket bookings
3. You will see a list of users, including passengers and bus operators.
4. For each user, you can:
   - **View Profile**: Access user details such as name, email, and phone number.
   - **Modify Booking**: Make changes to booking details in case of errors or complaints.
   - **Cancel Booking**: Cancel a booking upon customer request or system issue.

Manage Schedules

1. Go to the **Schedule Management** section.
2. View a list of all bus schedules, including routes, times, and operators.
3. To modify schedules:
   - **Create Schedule**: Add a new bus schedule with route details, departure times, and ticket prices.
   - **Edit Schedule**: Update existing schedules for changes in timing or routes.
   - **Delete Schedule**: Remove outdated or invalid schedules from the system.

Track Buses

1. Open the **Tracking Dashboard**.
2. View the live location of all active buses on an integrated map.

Revenue Tracking

1. Open the **Main Dashboard**.
2. Select a time frame.
3. View revenue data, including total income and number of tickets sold.

Feedback

Database Deployment

Feedback: The supervisor recommended deploying the database to a dedicated server to ensure better scalability, reliability, and accessibility for the system.

Action Taken: Explored options for cloud-based database hosting to improve performance and scalability. Plans are underway to migrate the database to a secure server environment.

Battery Optimization

Feedback: Clients complained about too much battery drainage.

Action Taken: Introduced configurable time intervals for location updates, with a default of 2 seconds, and provided a note about battery impact.

Ticket Verification

Feedback: Clients emphasized the importance of quick and accurate ticket scanning to avoid potential fake tickets

Action Taken: Enhanced QR code scanning performance and ensured real-time validation against the schedule and bus details.

Quick Tickets

Feedback: Clients requested a feature to allow passengers to purchase tickets even after the regular booking period had ended, such as when the bus is already en route.

Action Taken: Introduced the Quick Ticket service, enabling passengers to book tickets without selecting seats for buses that are within 24 hours of departure or already in transit.

Challenges

Real-Time Location Tracking

Challenge: Ensuring accurate and efficient real-time location updates while minimizing battery usage.

Solution: Introduced configurable tracking intervals to balance accuracy and battery life. Default intervals were set at 2 seconds, with options for longer durations, accompanied by clear user instructions.

Ticket Validation Accuracy

Challenge: Ensuring that scanned tickets are correctly validated against schedules, bus assignments, and status in real-time.

Solution: Enhanced the backend logic to cross-verify ticket details against multiple parameters, such as schedule, bus registration, and ticket validity, ensuring minimal delay and high accuracy.

Authentication Security

Challenge: Providing secure, user-friendly authentication options to meet diverse client needs.

Solution: Integrated Google Authentication for convenience and an in-house OTP system for added security, ensuring flexibility for different user preferences.

Performance Optimization

Challenge: Handling increased data requests efficiently, especially for real-time operations like location tracking and ticket validation.

Solution: Transitioned from the Google Distance Matrix API to the Directions API for faster performance. Additionally, stored frequently accessed backend data locally in files to reduce repeated API calls.

Usability and Accessibility

Challenge: Designing an intuitive interface for diverse users, including passengers, bus operators, bus owners and administrators.

Solution: Modernized UI features and libraries were used. Descriptive error messages are used to enhance usability.

Team Coordination

Challenge: Aligning tasks among team members, especially when addressing overlapping features or dependencies.

Solution: Used GitHub to track tasks, assign responsibilities, and ensure effective collaboration. Regular team meetings helped address issues.

System Scalability

Challenge: Designing a system that can handle a growing user base and expanding operational requirements.

Solution: Implemented scalable backend architecture, optimized database queries, and tested for high user loads to ensure smooth performance under increased demand.

Work Task Division and Individual Contribution

Task		Contribution
E/20/032	Admin site and database	Developed the admin portal interface, implemented features for schedule management and revenue tracking, and designed the database schema.
E/20/035	Mobile app	Built the mobile app front-end, integrated Google Auth and OTP authentication, implemented ticket booking and scanning features etc., and connected the app to the backend.
E/20/133	Backend and database	Developed backend APIs for ticket booking, scanning, and location tracking, transitioned to the Directions API for improved performance, and managed database connectivity and optimizations.
E/20/157	WebApp, Backend and database	Designed and developed the website interface, with Google Auth, OTP authentication, ticket booking, and schedule features; created backend APIs for booking and tracking; optimized database connectivity and integrated the Directions API.

Declaration

We, the undersigned, hereby declare that the information presented in this report is accurate and truthful to the best of our knowledge and understanding. All work described herein has been carried out by the group members as outlined, and any external contributions or resources have been appropriately acknowledged.

Registration Number	Name	Signature
E/20/032	Bandara A.M.N.C.	[illegible]
E/20/035	Bandara K.C.H.N.A.W.M.R.C.J.N.	[illegible]
E/20/133	Haththella H.A.D.T.N.	[illegible]
E/20/157	Janakantha S.M.B.G.	[illegible]