12306LiteX: A Simplified 12306 train ticket booking platform

CS-GY 9053, Java Final Project Report, Fall 2024

Dec. 16

Yuheng Wu, netID: yw5372

Xingyu Han, NetID: xh2787

Weizhen Zhou, NetID: wz3008

Our Github Reposiitory Link

Overview

- 12306 is the official online ticket booking platform for China's railway system, operated by China Railway Corporation. It's one of the most widely used systems in China, serving millions of users daily, especially during peak travel seasons like the Chinese New Year, where demand for tickets skyrockets. The platform enables users to book tickets, check train schedules, manage reservations, and handle refunds.
- Our goal is to build a simplified ticket booking system inspired by the 12306 platform, allowing the function:

1. register and login
2. administrator login
3. change password and edit user profile
4. search for train schedules by departure station, arrival station and date
5. check train information including stop stations, seats available and name
6. book ticket, pay for ticket and cancel ticket.

- The system will be developed in Java using Spring to leverage modern frameworks for database management, efficient processing, and a user-friendly interface.
- As for innovation, we plan to add an AI-based module to guide user through jumping to different asked panels as well as give tunned suggestions.

Features

- GUI (JavaFX)
- [Not Java threads] Threads (SpringBoot Tomcat & Hikari)
- Database (MySQL)
- SpringBoot Framework
- Java RESTful style APIs and Web Service (HTTP request)
- JDBC
- Sychronization ReentrantLock (Atomic book operation)

- Security (AES Password Encryption)

How To Run (Prefer using Jetbrains IDEA)

- Run your MySQL server, note its port and location
- Run `create database javaproject` in MySQL console
- First run `database/MySQL_DDL.sql` , then run `database/mysql_dml.sql`
- Config `SpringBackend/src/main/resources/application.properties`

  The current `tomcat` and `hikari` config works well on `Intel Core i9-14900HX`

  ```
  spring.application.name=SpringBackend

  spring.datasource.url=//replace with database url
  spring.datasource.username=//replace with database username
  spring.datasource.password=//replace with database password
  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
  spring.jpa.show-sql=true
  spring.jpa.properties.hibernate.format_sql=true
  spring.jpa.hibernate.ddl-auto=update
  server.port=8088

  server.tomcat.threads.max=300
  server.tomcat.threads.min-spare=30
  server.tomcat.accept-count=200

  spring.datasource.hikari.maximum-pool-size=30
  spring.datasource.hikari.connection-timeout=20000
  spring.datasource.hikari.minimum-idle=10

  langchain4j.open-ai.chat-model.api-key=${OPENAI_KEY}//replace with OpenAI token
  langchain4j.open-ai.chat-model.model-name=gpt-4o-mini
  langchain4j.open-ai.chat-model.log-requests=true
  langchain4j.open-ai.chat-model.log-responses=true
  ```

- Make sure IDEA recognize and load `Maven` correctly.
- If you want to change resolution, config `JavaFXFrontend/src/main/java/com/wxy/javafxfrontend/Settings.java`, and change `res_width` and `res_height`. (Current `1920*1080` works on `3840*2160` screen)
- Run `SpringBackend/src/main/java/com/wxy/springbackend/SpringBackendApplication.java`
- Run `JavaFXFrontend/src/main/java/com/wxy/javafxfrontend/JavaFXApplication.java`
- NOTICE: There may be latency in OpenAI APIs response.

Basic Function

User Interface

- User login: User can login with different accounts. Each account holds user's information and user's ticket information.
- Administrator login: User can login as an administrator to view the information of platform including train map and train schedule.
- User information lookup and edit: User can update their information and search for their bought tickets.
- Password change: User can change their password with old password check.

Ticket Interface

- Ticket consult: User can consult train's combination from departure location to arrival location, provided with date and other sorting options. There are other small features like switch departure to arrival button, system time display, etc.
- Ticket consult result: This panel shows the result of the consult, and the result can be switched between different days, and each result listed should include informations like departure, arrival, time, time spent, price, ticket left(different classes).
- Ticket booking and details: Clicking on one consult result will shift user to another panel that shows the details of the train choosed and let user choose the class they want(First class seat or second class seat).
- Ticket booking: After choose the specific routine with seat class, user can now choose whether to buy the ticket with a 10 min time limit, shown on the corner. If the time limit passed, the buying process will be canceled. The booking process is atomic, which grasps the ticket first.
- Ticket payment: User can pay for their ticket right after booking or later in the Panel 'MyOrder' where they can view all of their tickets.
- Ticket cancel: User can cancel ticket before the departure time.

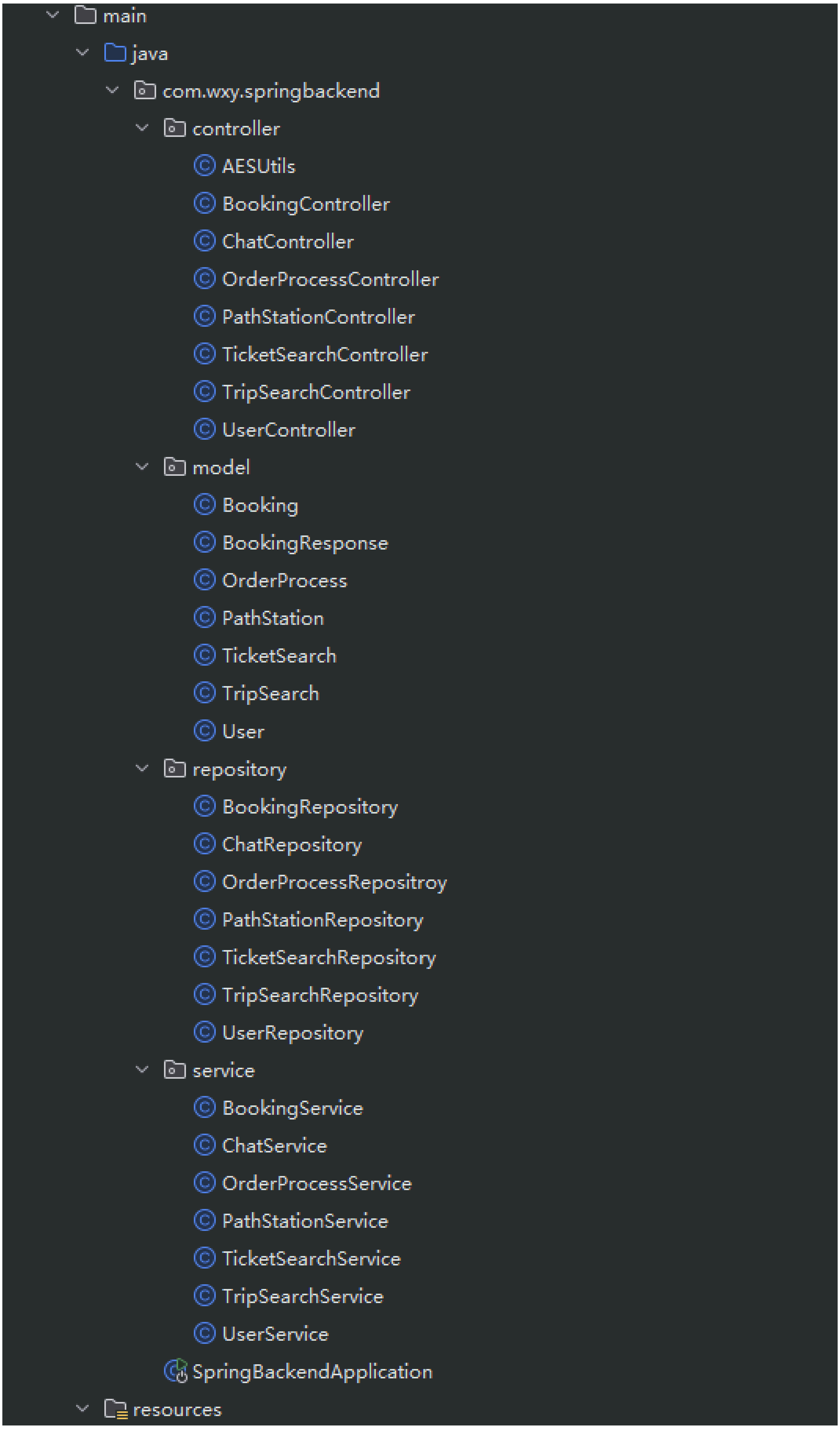
AI (with OpenAI APIs) Interface

We aim to develop an AI-based assistant within the app that can guide users efficiently, inspired by the customer support model in apps like Bank of America.

- User Interaction Panel: Users can click on a chat interface and type in their requests or questions. This could range from asking how to book a ticket to changing their password.
- Natural Language Processing: The AI assistant interprets the user's input by processing it through an AI API, transforming the user's request into actionable commands for the app.
- Detailed Guidance: The assistant will provide step-by-step instructions to guide users through the requested processes.
- Direct Navigation Links: After generating guidance, the AI assistant will offer a direct link within the chat interface. When clicked, this link will take users directly to the relevant feature page or function within the app.
- Feature Availability Feedback: If a requested feature is not yet supported in the app, the AI assistant will inform the user clearly.

- Contextual Replies for Off-Topic Queries: If the user’s input is unrelated to app services, the AI assistant will respond with a polite message to tell the user.

Backend



```
main
  java
    com.wxy.springbackend
      controller
        AESUtils
        BookingController
        ChatController
        OrderProcessController
        PathStationController
        TicketSearchController
        TripSearchController
        UserController
      model
        Booking
        BookingResponse
        OrderProcess
        PathStation
        TicketSearch
        TripSearch
        User
      repository
        BookingRepository
        ChatRepository
        OrderProcessRepositroy
        PathStationRepository
        TicketSearchRepository
        TripSearchRepository
        UserRepository
      service
        BookingService
        ChatService
        OrderProcessService
        PathStationService
        TicketSearchService
        TripSearchService
        UserService
      SpringBackendApplication
  resources
```

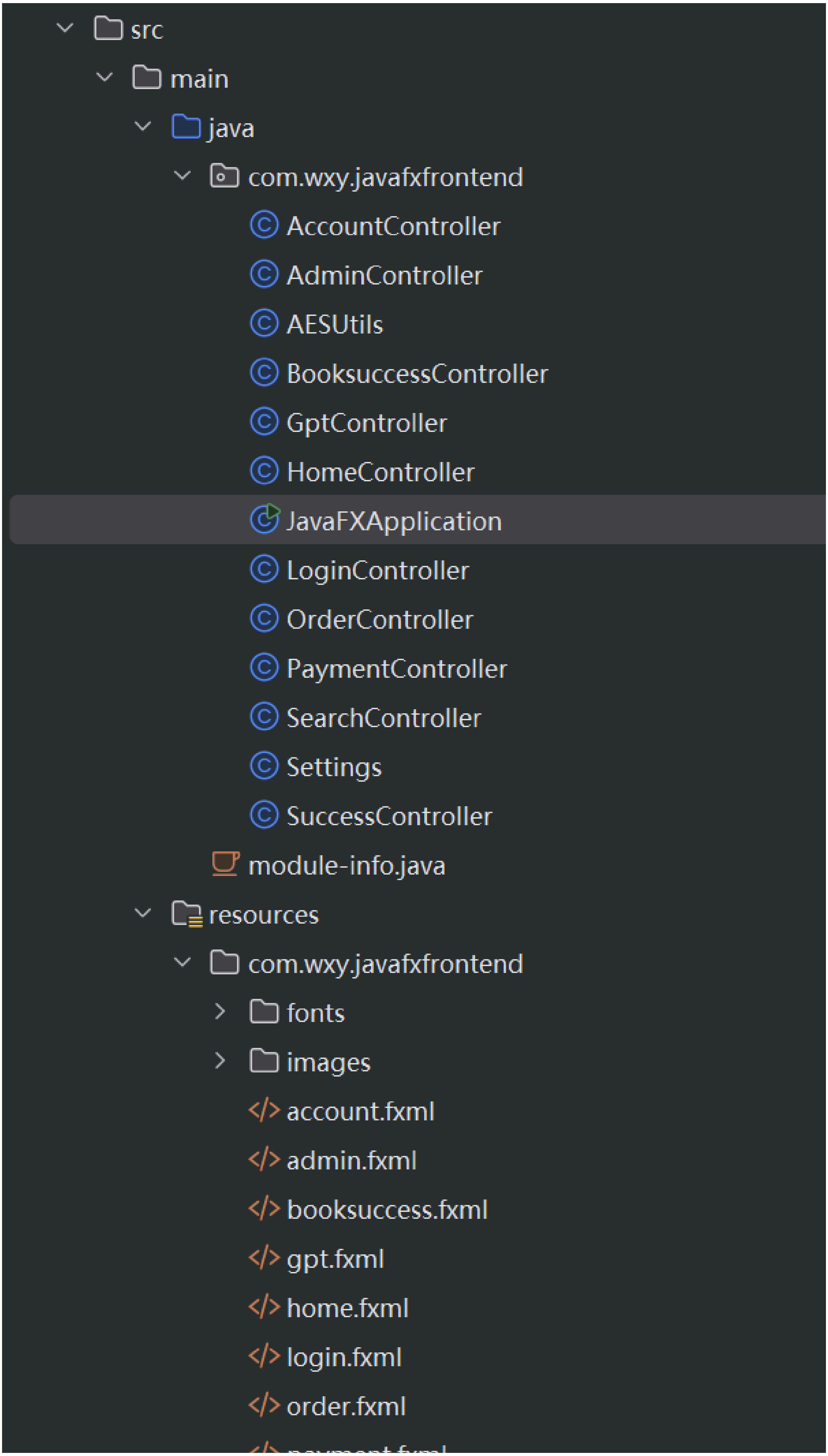
For backend, we use Java RESTful APIs for web service and JDBC APIs for database query and update.

We create multiple `Controller`, `Repository` and `Service` entity to implement the function, with `Controller` for API address and HTTP function mapping, `Service` for function call and required process, `Repository` for database connection, utilizing JDBC APIs.

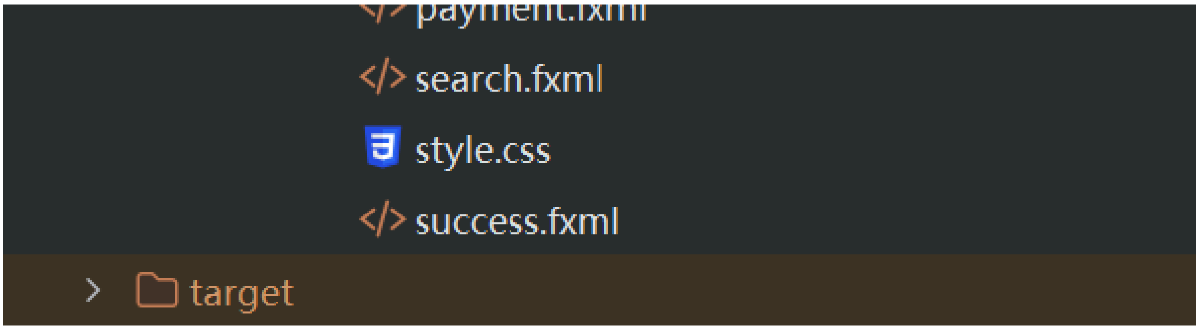
We create `DTO` class in folder `model` to normalize data transmission, we use Java Springboot Jdbc features and prepareStatements to replace our DTO class variables to fill the "?" in our SQL String, so there is no possibility to attacked by SQL injection.

We use oracle to design database schema, and MySQL for DDL and DML code.

Frontend



```
src
  main
    java
      com.wxy.javafxfrontend
        AccountController
        AdminController
        AESUtils
        BooksuccessController
        GptController
        HomeController
        JavaFXApplication
        LoginController
        OrderController
        PaymentController
        SearchController
        Settings
        SuccessController
      module-info.java
    resources
      com.wxy.javafxfrontend
        fonts
        images
        account.fxml
        admin.fxml
        booksuccess.fxml
        gpt.fxml
        home.fxml
        login.fxml
        order.fxml
        payment.fxml
```



payment.fxml
search.fxml
style.css
success.fxml
target

We use JavaFX as our frontend instead of Java Swing because its modern feature and compatiblity with CSS. Each `.fxml` coordinates with a Controller. `Settings.java` is used to adjust resolution. Frontend has same static DTO class and use `ObjectMapper` to map received response to a class object. Frontend use `HttpClient` to send and receive HTTP requests.

We use a overall `style.css` to render display items.

AI Interface

Prompt Engineering

We employ LangChain for Java as the foundational framework to interface with LLMs. Our approach leverages prompt tuning to ensure the model consistently understands its role as a train ticket subscription assistant, and adheres to specified behavioral guidelines.

Structured Response

The model's responses must follow a strict JSON format to machine-readable outputs. For example, the model might produce:

```
{
    "message": "Ah, Florida sounds tempting! But tonight might be a bit of a stretch unless you have a magic carpet. Let's check the trains first!",
    "Instruction": "ShowTrains",
    "Param1": "New York",
    "Param2": "Florida",
    "Param3": "2024-12-16"
}
```

- message: A user-facing message displayed in the frontend UI.

- Instruction & Params: The chosen action (e.g., `ShowTrains`) and up to three parameters needed to execute that action.

Any guildence or casual conversation appears exclusively in the `message` field, while operational instructions remain separated and easily parsed by the application that the frontend can excute the redirection logic based on these.

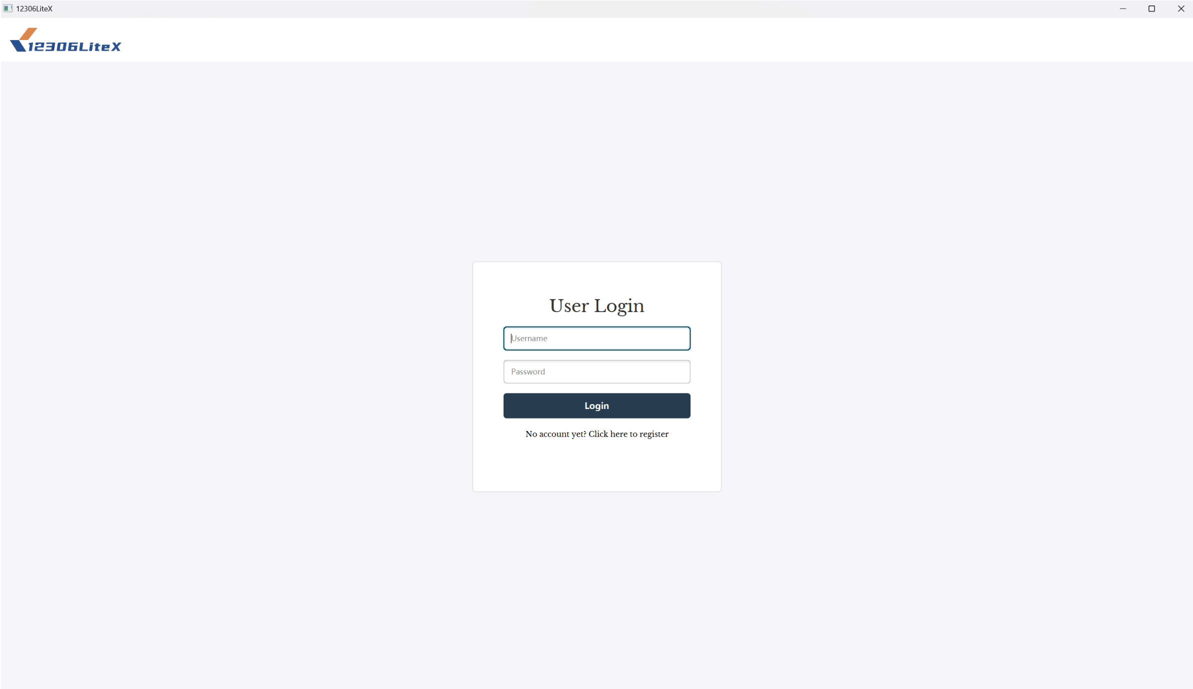
Function Calling

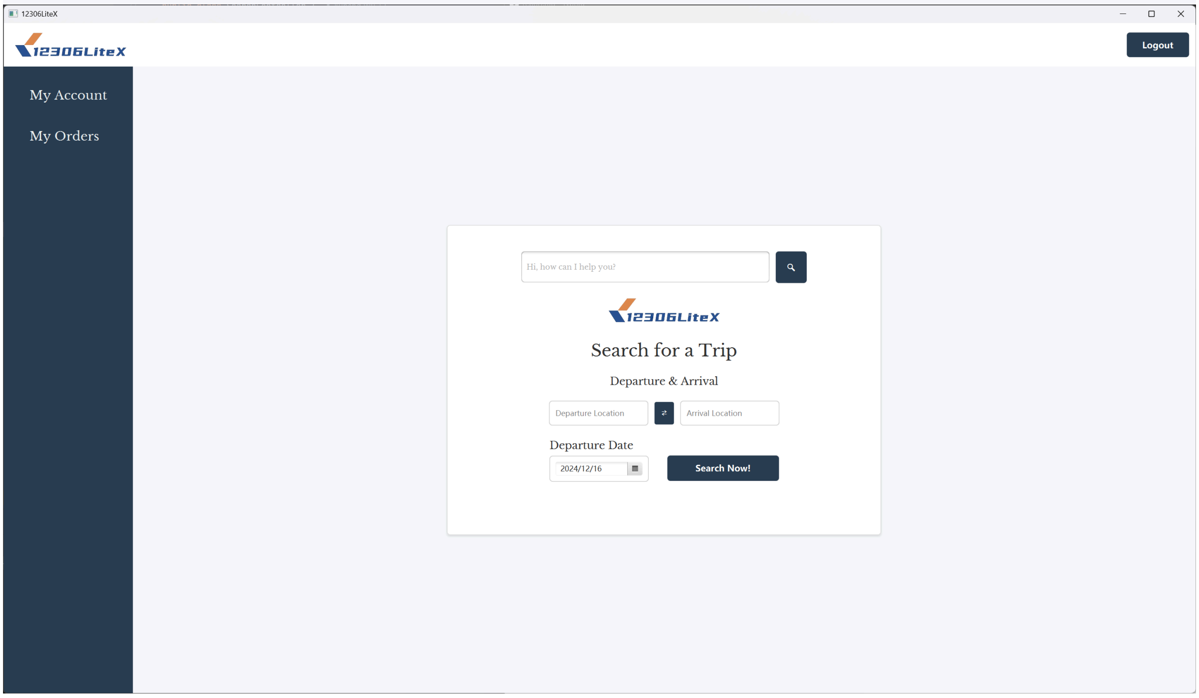
We integrate a function calling layer to enrich the model's responses with dynamic, real-time data. For example, we may have a function `getCurrentLocation()` or `getCurrentDate()` that the model can produce more grounded replies and tailor its instructions dynamically based on real-time data..

Rollback Mechanism

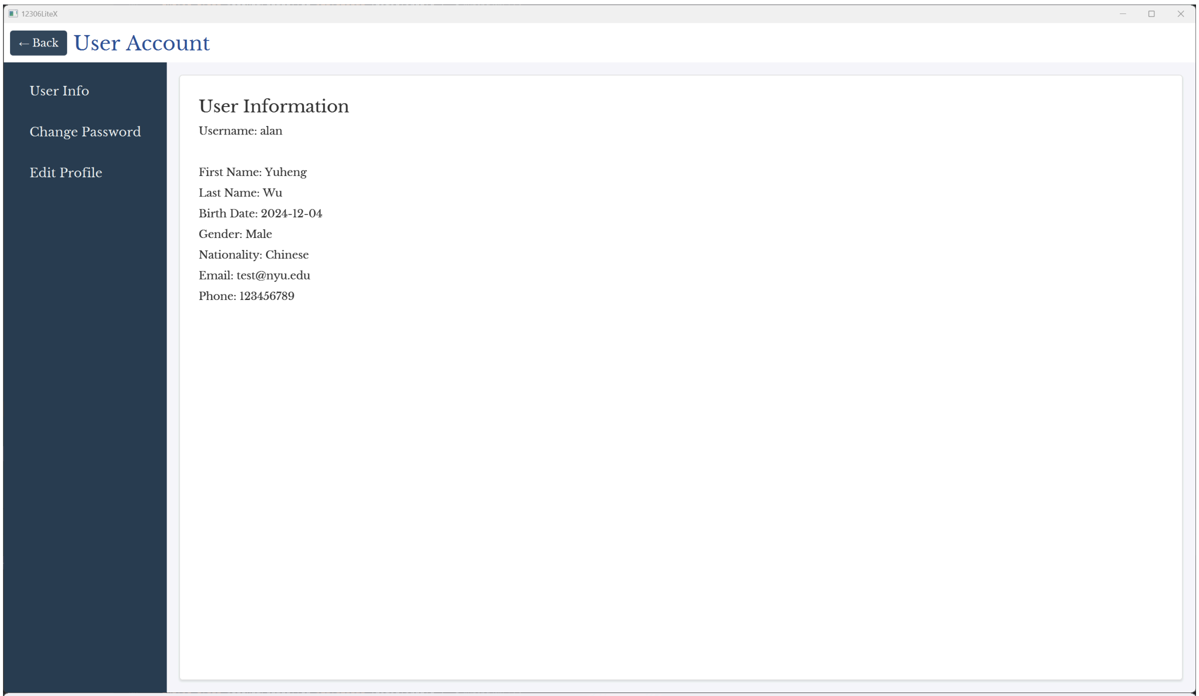
If the model’s response is invalid, incomplete, or does not follow the specified structure, we implement a rollback mechanism. The last user message and any relevant context are resent to the model, prompting it to regenerate its response according to the strict output format.

GUI Preview





2024/12/16



User Information

Username: alan

First Name: Yuheng

Last Name: Wu

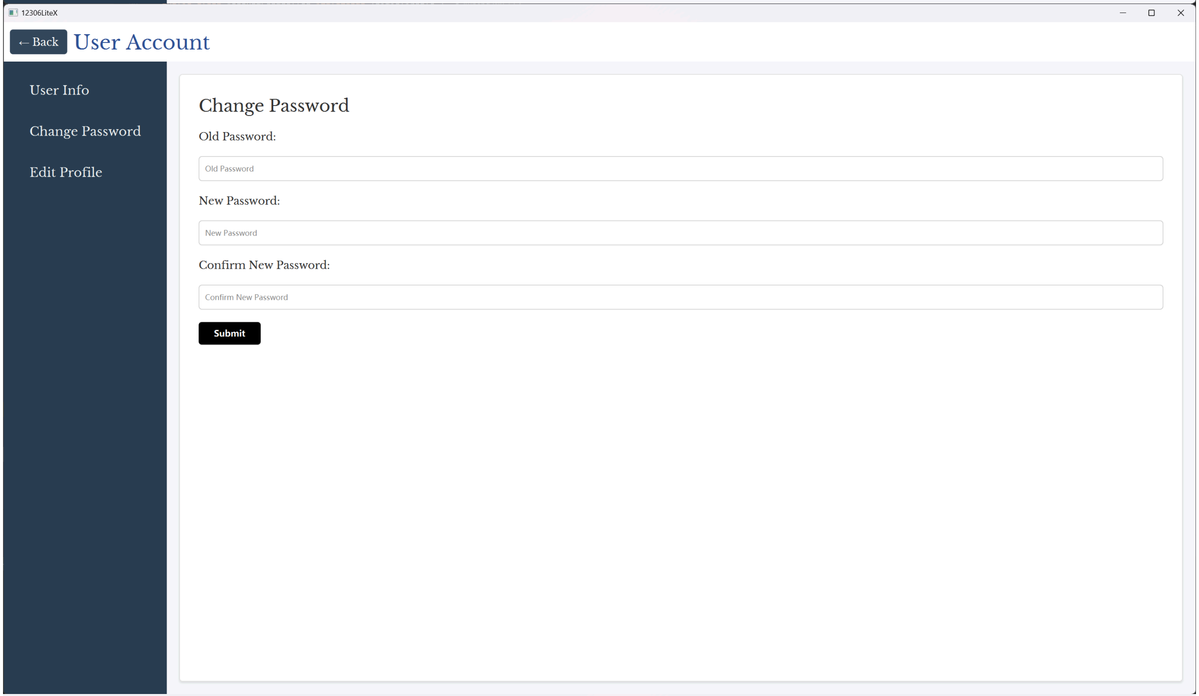
Birth Date: 2024-12-04

Gender: Male

Nationality: Chinese

Email: test@nyu.edu

Phone: 123456789

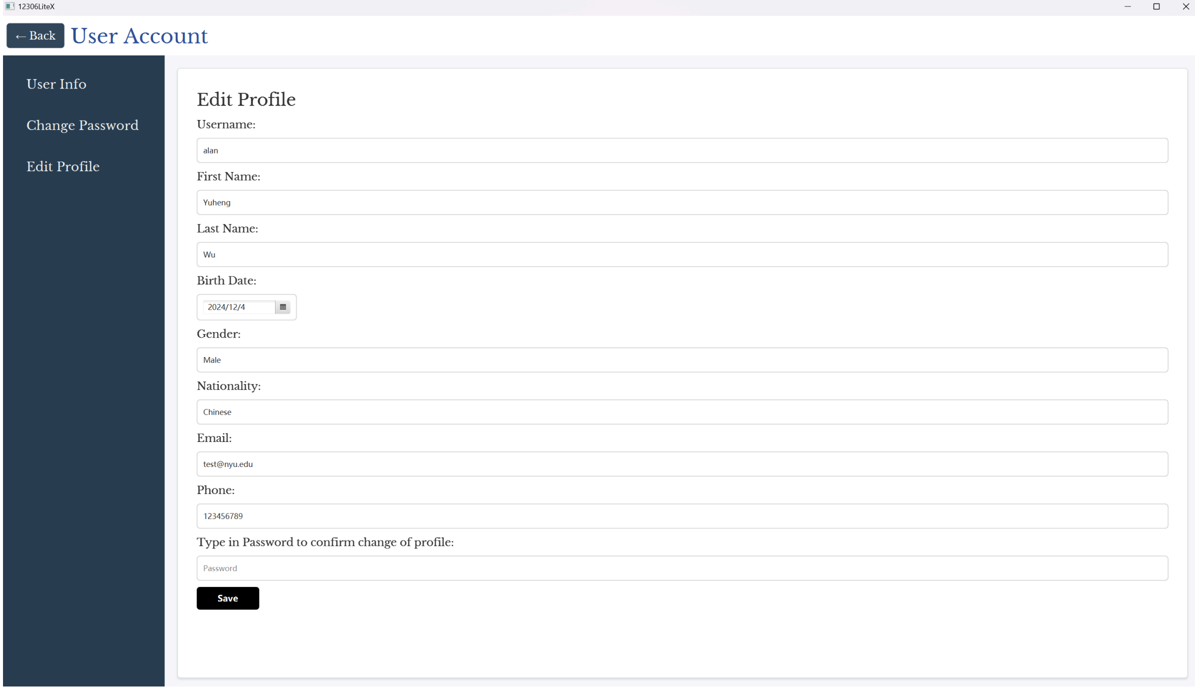


Change Password

Old Password: Old Password

New Password: New Password

Confirm New Password: Confirm New Password



Edit Profile

Username: alan

First Name: Yuheng

Last Name: Wu

Birth Date: 2024/12/4

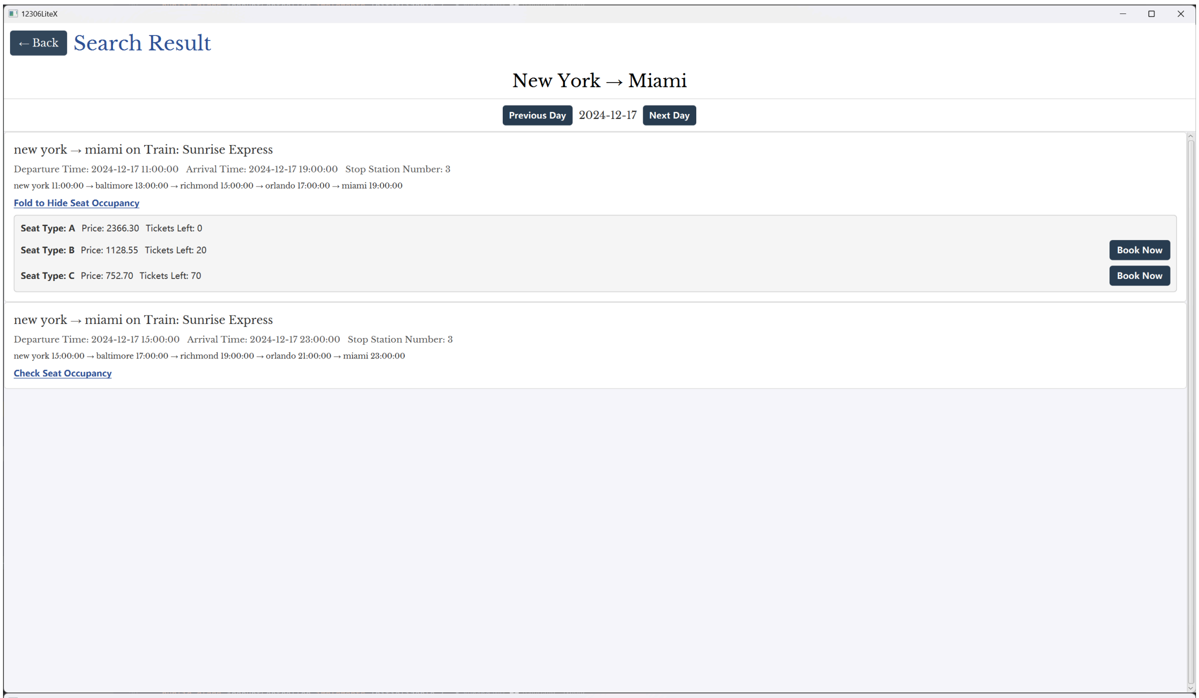
Gender: Male

Nationality: Chinese

Email: test@nyu.edu

Phone: 123456789

Type in Password to confirm change of profile: Password



Search Result

New York → Miami

2024-12-17

new york → miami on Train: Sunrise Express

Departure Time: 2024-12-17 11:00:00 Arrival Time: 2024-12-17 19:00:00 Stop Station Number: 3

new york 11:00:00 → baltimore 13:00:00 → richmond 15:00:00 → orlando 17:00:00 → miami 19:00:00

Fold to Hide Seat Occupancy

Seat Type: A Price: 2366.30 Tickets Left: 0

Seat Type: B Price: 1128.55 Tickets Left: 20

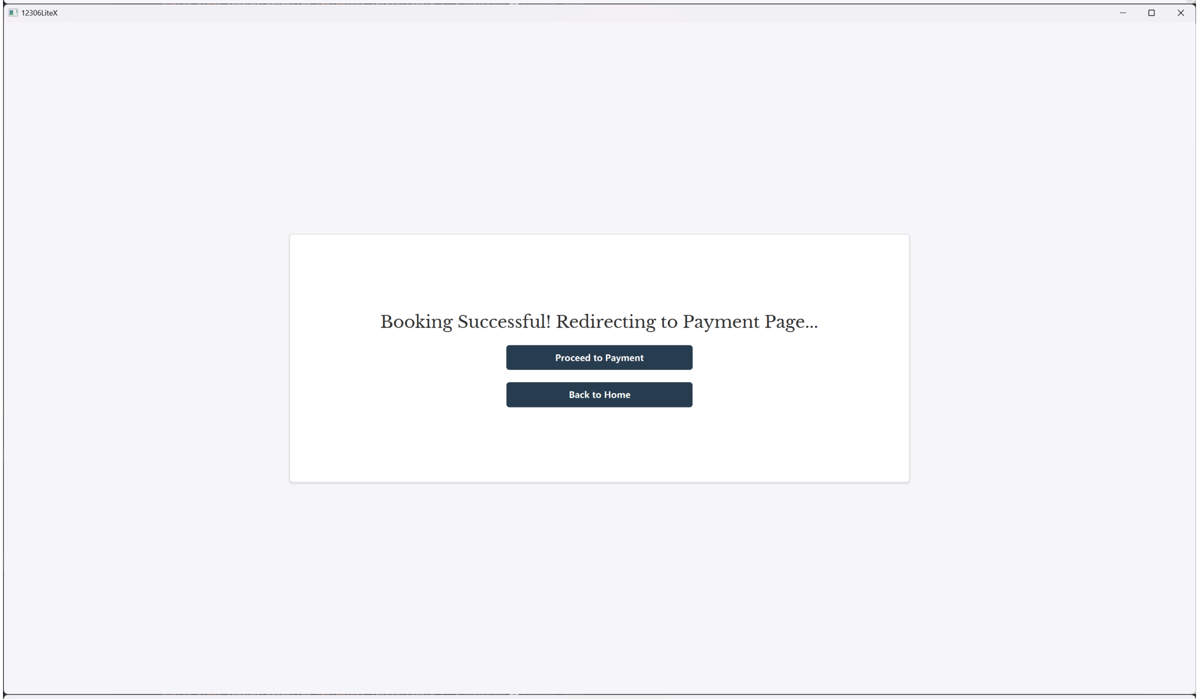
Seat Type: C Price: 752.70 Tickets Left: 70

new york → miami on Train: Sunrise Express

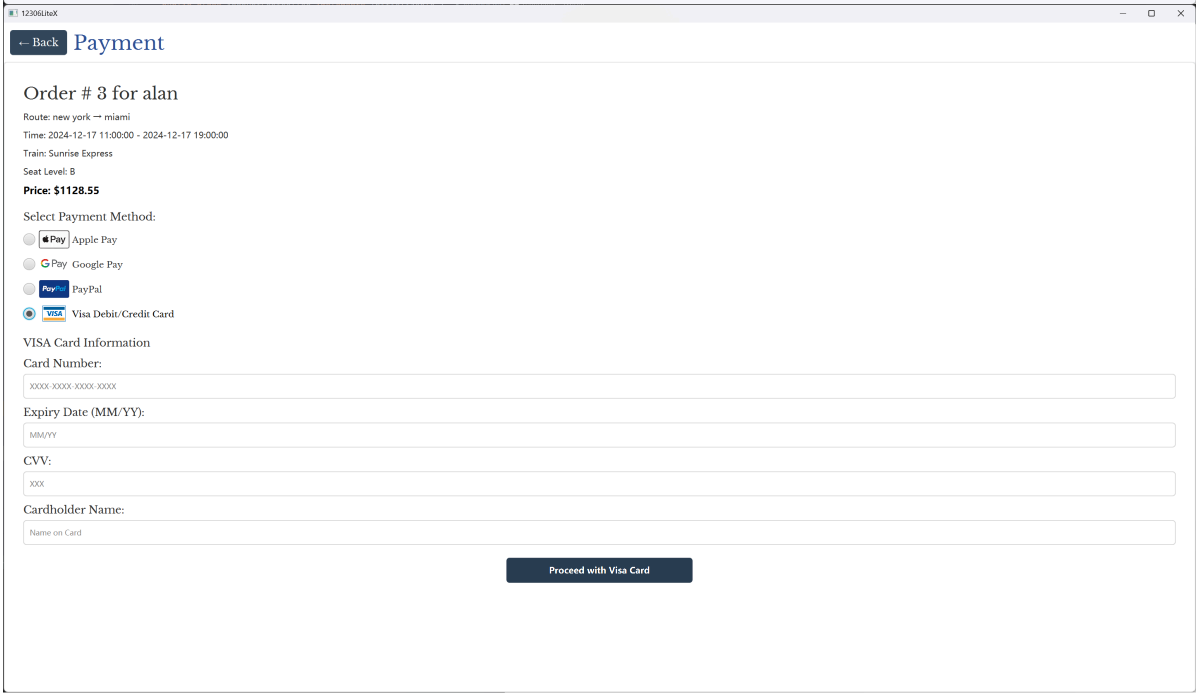
Departure Time: 2024-12-17 15:00:00 Arrival Time: 2024-12-17 23:00:00 Stop Station Number: 3

new york 15:00:00 → baltimore 17:00:00 → richmond 19:00:00 → orlando 21:00:00 → miami 23:00:00

Check Seat Occupancy



Booking Successful! Redirecting to Payment Page...



Payment

Order # 3 for alan

Route: new york → miami

Time: 2024-12-17 11:00:00 - 2024-12-17 19:00:00

Train: Sunrise Express

Seat Level: B

Price: $1128.55

Select Payment Method:

○ Apple Pay

○ Google Pay

○ PayPal

● Visa Debit/Credit Card

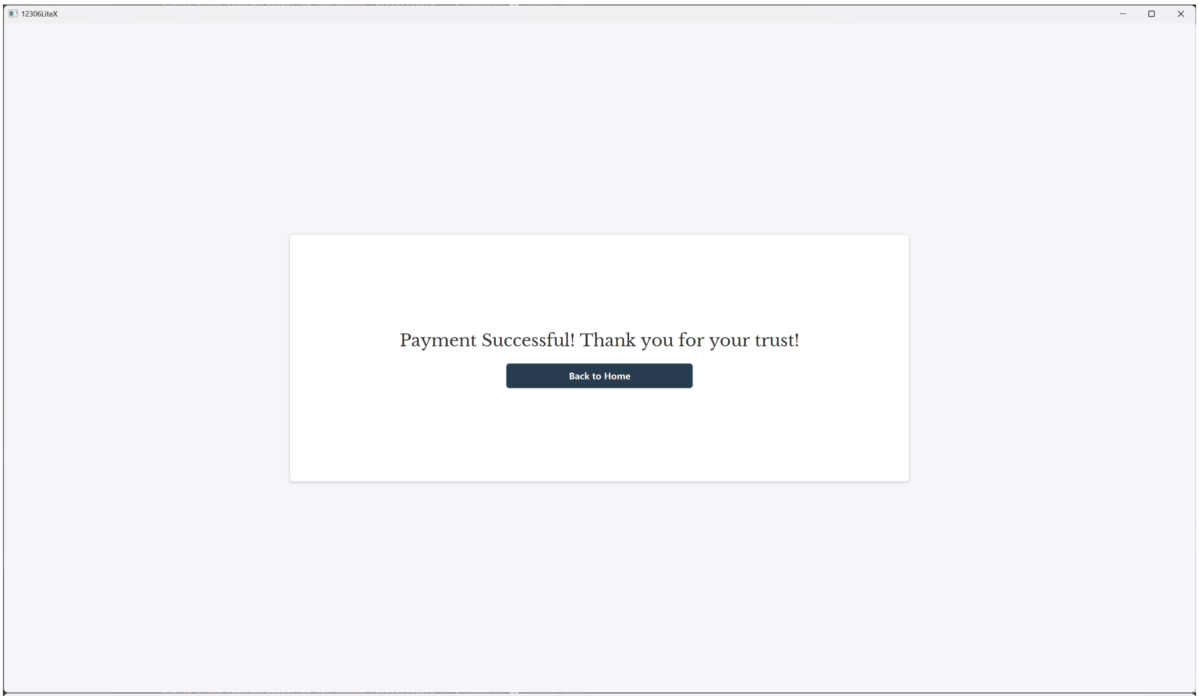
VISA Card Information

Card Number:

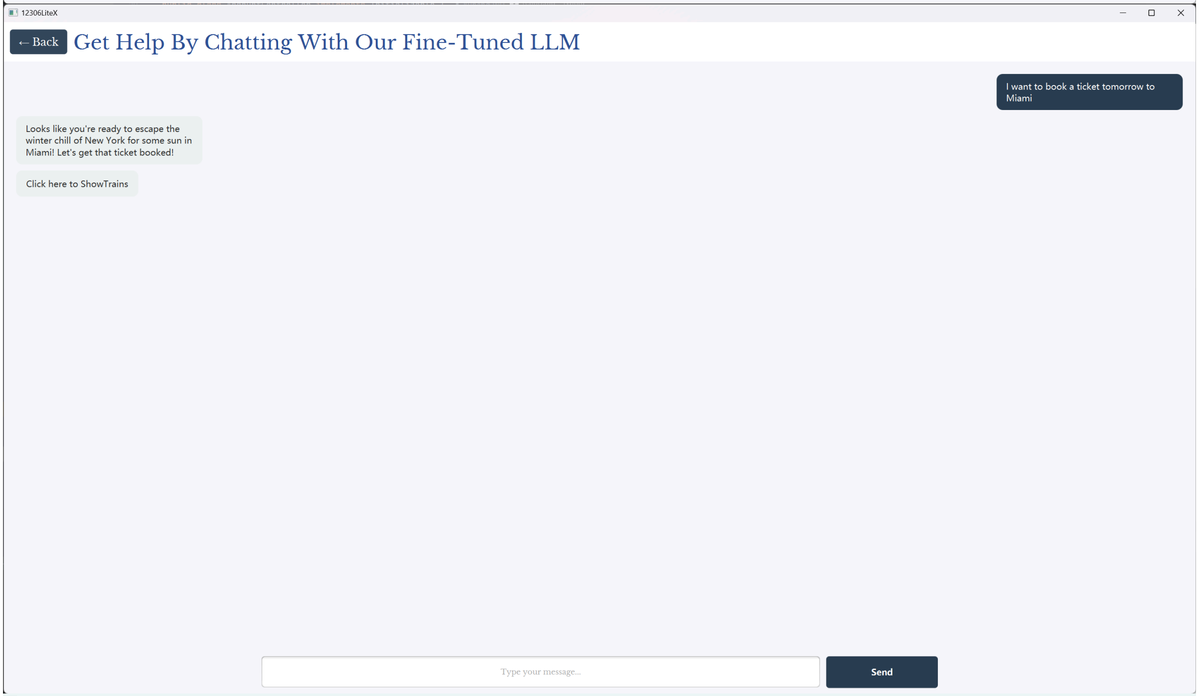
Expiry Date (MM/YY):

CVV:

Cardholder Name:



Payment Successful! Thank you for your trust!

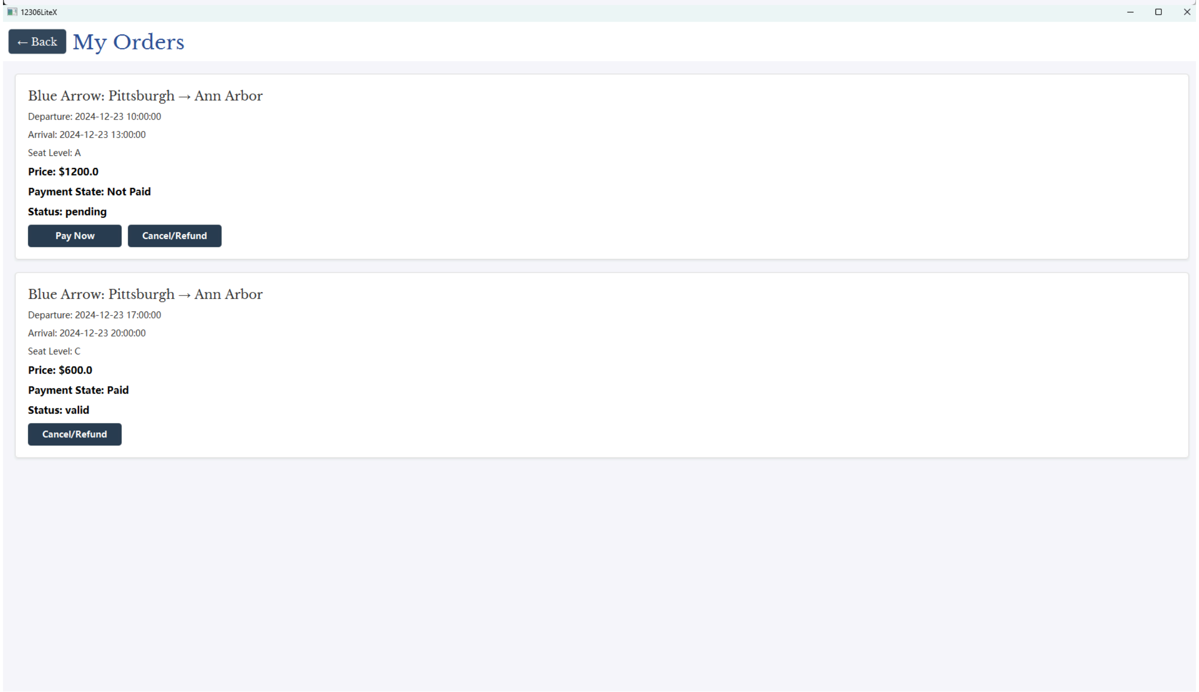


Get Help By Chatting With Our Fine-Tuned LLM

I want to book a ticket tomorrow to Miami

Looks like you're ready to escape the winter chill of New York for some sun in Miami! Let's get that ticket booked!

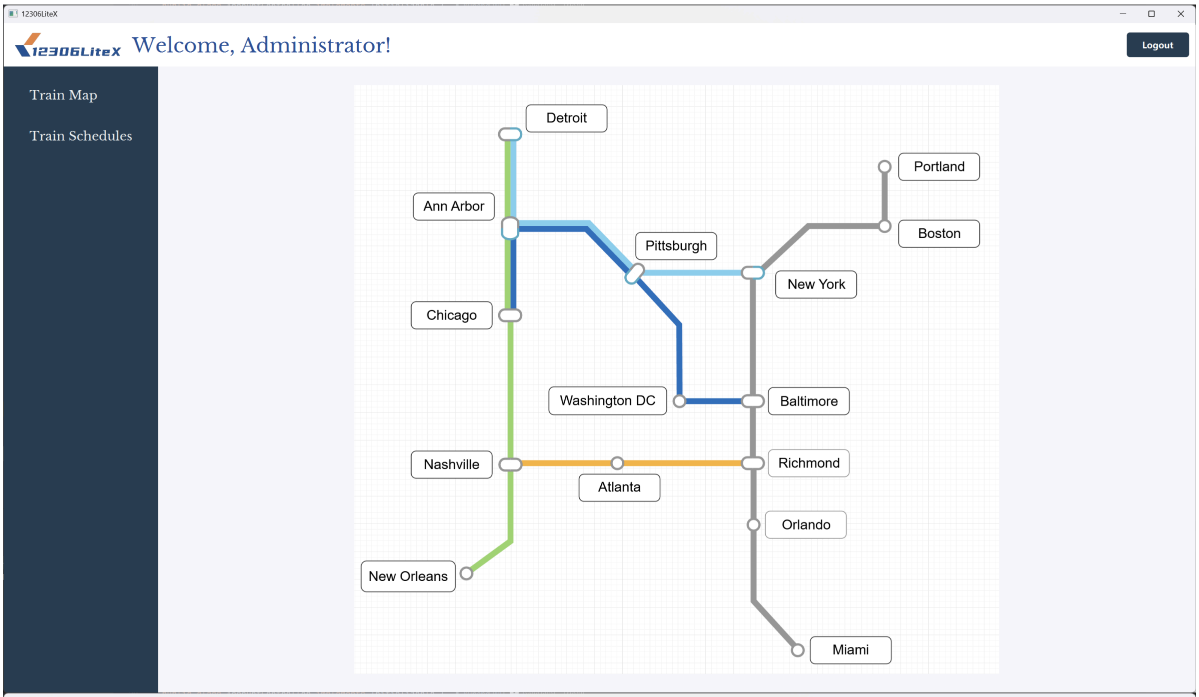
Click here to ShowTrains

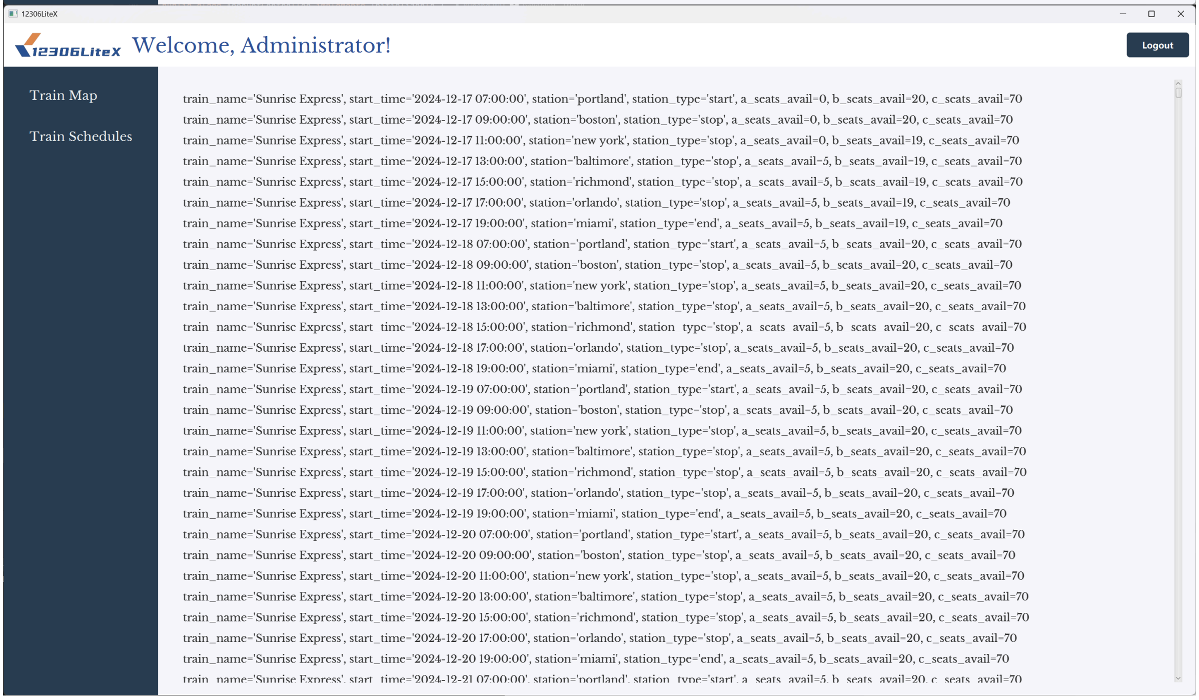


My Orders

Blue Arrow: Pittsburgh → Ann Arbor
Departure: 2024-12-23 10:00:00
Arrival: 2024-12-23 13:00:00
Seat Level: A
Price: $1200.0
Payment State: Not Paid
Status: pending

Blue Arrow: Pittsburgh → Ann Arbor
Departure: 2024-12-23 17:00:00
Arrival: 2024-12-23 20:00:00
Seat Level: C
Price: $600.0
Payment State: Paid
Status: valid





```
train_name='Sunrise Express', start_time='2024-12-17 07:00:00', station='portland', station_type='start', a_seats_avail=0, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-17 09:00:00', station='boston', station_type='stop', a_seats_avail=0, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-17 11:00:00', station='new york', station_type='stop', a_seats_avail=0, b_seats_avail=19, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-17 13:00:00', station='baltimore', station_type='stop', a_seats_avail=5, b_seats_avail=19, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-17 15:00:00', station='richmond', station_type='stop', a_seats_avail=5, b_seats_avail=19, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-17 17:00:00', station='orlando', station_type='stop', a_seats_avail=5, b_seats_avail=19, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-17 19:00:00', station='miami', station_type='end', a_seats_avail=5, b_seats_avail=19, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 07:00:00', station='portland', station_type='start', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 09:00:00', station='boston', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 11:00:00', station='new york', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 13:00:00', station='baltimore', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 15:00:00', station='richmond', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 17:00:00', station='orlando', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-18 19:00:00', station='miami', station_type='end', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 07:00:00', station='portland', station_type='start', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 09:00:00', station='boston', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 11:00:00', station='new york', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 13:00:00', station='baltimore', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 15:00:00', station='richmond', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 17:00:00', station='orlando', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-19 19:00:00', station='miami', station_type='end', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 07:00:00', station='portland', station_type='start', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 09:00:00', station='boston', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 11:00:00', station='new york', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 13:00:00', station='baltimore', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 15:00:00', station='richmond', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 17:00:00', station='orlando', station_type='stop', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-20 19:00:00', station='miami', station_type='end', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
train_name='Sunrise Express', start_time='2024-12-21 07:00:00', station='portland', station_type='start', a_seats_avail=5, b_seats_avail=20, c_seats_avail=70
```