DESIGN OF A TRAVEL ITENERARY PLANNING AND RESERVATION SYSTEM

By

GROUP ID: 29

Yashkumar Paneliya (22CS60R70)

Shailesh Chaudhary (22CS60R37)

Submitted to

Prof. Pallab Dasgupta

Prof. Sourangshu Bhattacharya

COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

TABLE OF CONTENTS

LIST OF FIGURES

Introduction:

- Travel itinerary and reservation system is a portal or application where a user can provide several places that he/she want to visit.
- In addition to that several constraints like arrival date at a particular place, number of days to stay at a particular place, mode of transportation, etc.
- On the basis of these inputs, the system generates an itinerary plan along with status of tickets for suggested travel mode between consecutive places.
- Once the plan is generated, user is given an option to confirm their booking and after successful payment, the bookings are done and stored in user's profile.

Features:

- Authentication and registration
- Itinerary planning
- Booking tickets
- Payment

Use cases:

- Use Cases describe the sequence of events that an actor take place using the system in order to complete a process.
- Use case identifies the actors and the processes they perform.
- In the following table all the use cases that may be included in this project is described.

Use case	Actor
Registration	Traveller (User)
Login	
Input places	
Itinerary generation	
Confirm bookings	
View bookings	
Payment	

Use case: Registration

- Actor: Traveller
- Description:
    - Traveller first has to register to use the feature of itinerary planning in the system. Registration requires email, name and a password from the user.

- An HTTP request is then sent to server with all these data and server checks whether this user already exists or not, if no then registers the user and store the data into database and user is redirected to the home page of system.

Use case: Login

- Actor: Traveller
- Description:
  - If user has already registered to the system, then he/she can directly login to the system.
  - Login requires email and password
  - An HTTP request is sent to server for verification of credentials, if verified then user is redirected to home page and its username is stored in the local storage of the end-user machine for further interaction with server.

Use case: Input places

- Actor: Traveller
- Description:
  - User has choices to input three different places, their arrival dates, number of days to stay and mode of transportation.
  - Currently the choice of places is fixed.
  - Transportation modes can be flight, bus or train.
  - Several constraints are also imposed, places and dates should be different, difference between dates should not be smaller than number of days of stay between two places, etc.

Use case: Itinerary Planning

- Actor: Traveller
- Description:

- Once the input is provided, the data is sent to server as a POST request.
- Server processes the data and generates the itinerary plan along with ticket costs, status of availability of tickets and order of places to visit.
- Whole plan is sent to client and an option to confirm booking is provided.

Use case: Confirm booking

- Actor: Traveller
- Description:
  - An HTTP request is sent to server to confirm the booking.
  - Server then generates a unique ID for the plan and stores it into user's data and also the payment data is stored in separate data file of admin analysis.
  - After successful confirmation, user can view their bookings on system.

Use case: Payment

- Actor: Traveller
- Description:
  - Before confirming the bookings, user has to pay the total cost of tickets.
  - On successful payment, bookings get confirmed and database entries are modified (for tickets).

Use case: View booking

- Actor: Traveller
- Description:
  - A user can view his/her all bookings in the profile section in system.
  - Bookings are stored in separate files for each user and fetched according to logged in user.

Use case Diagram:

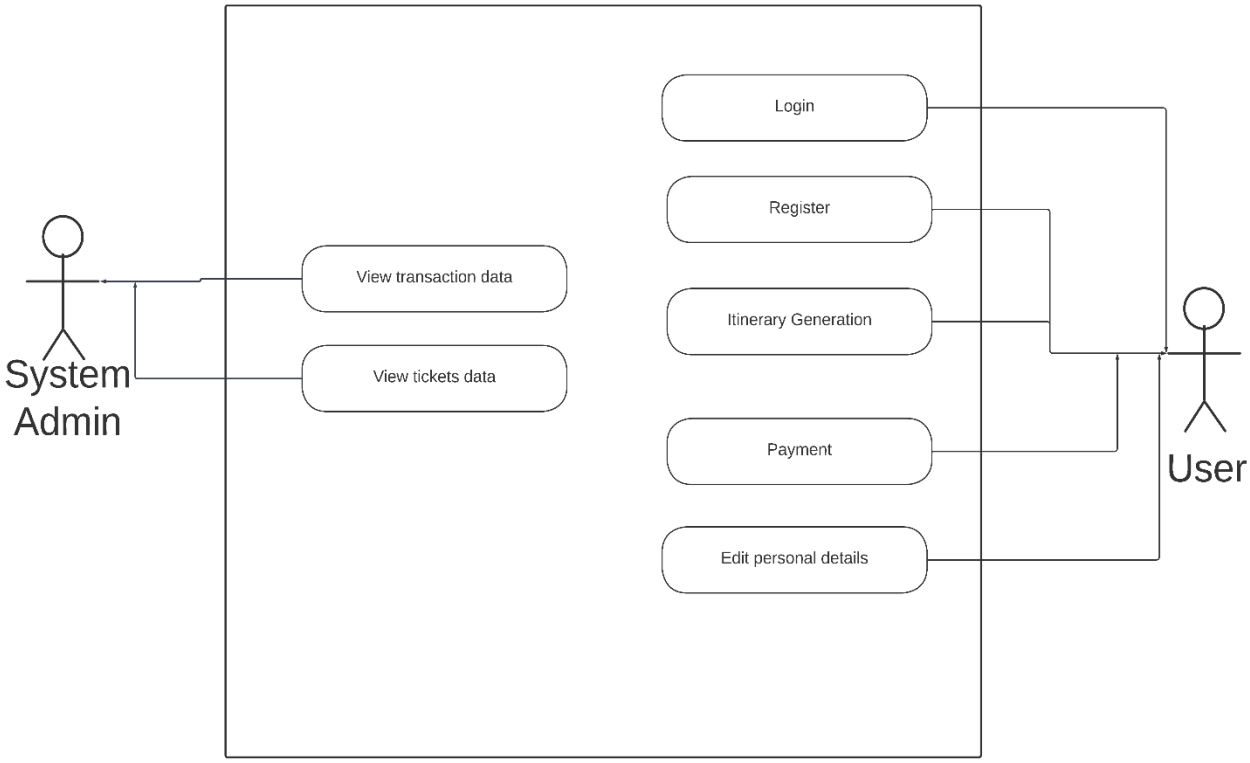

Fig 1: Use case diagram

Working flow:

- Server will be hosted on the defined port (8080 for testing purpose)
- Several paths are defined in the server for handling the requests for the respective path
    a. /: Home page
    b. /register: Register page
    c. /itenarary: Input form for places to travel with constraints
    d. /profile: User profile
- Several paths are defined for both GET and POST requests on same path.
- GET will simply serve the webpage according to path whereas POST request will send data to server to perform operations on it like generating itinerary.
- On providing the places data, a POST request will be sent to server with the data, server generates a plan with ticket status and all the costs.
- All the data regarding ticket counts and cost is stored on the server in CSV files.
- Once the plan is generated, it is sent to client as a response to POST request.
- User is prompted to pay for booking tickets.
- On successful payment, the plan is stored with a unique ID in both user profile and admin transactions.

Languages & Tools:

Frontend**:** HTML, JavaScript

Style**:** CSS

Backend**:** Python

Data storage: CSV files

IDE: VSCode

Operating System: Ubuntu Linux

External library: Pandas (For CRUD operations on datasheets)

Sequence Diagram:

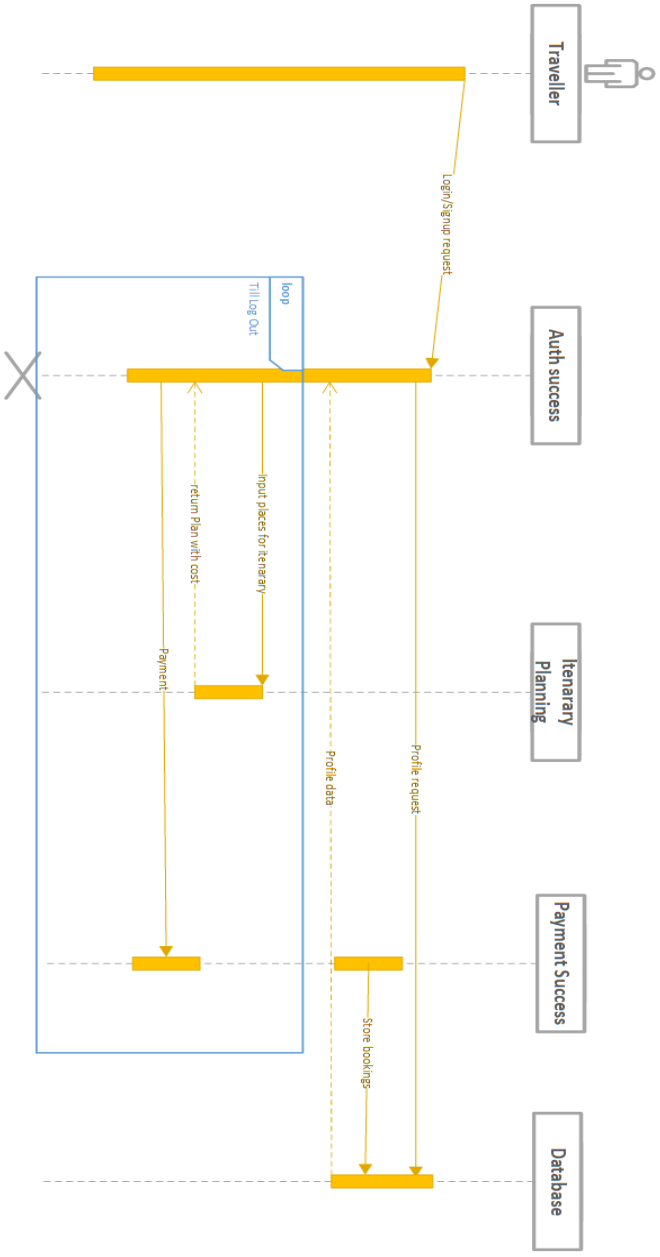

Fig 2: System Sequence Diagram

Implementation snapshots:

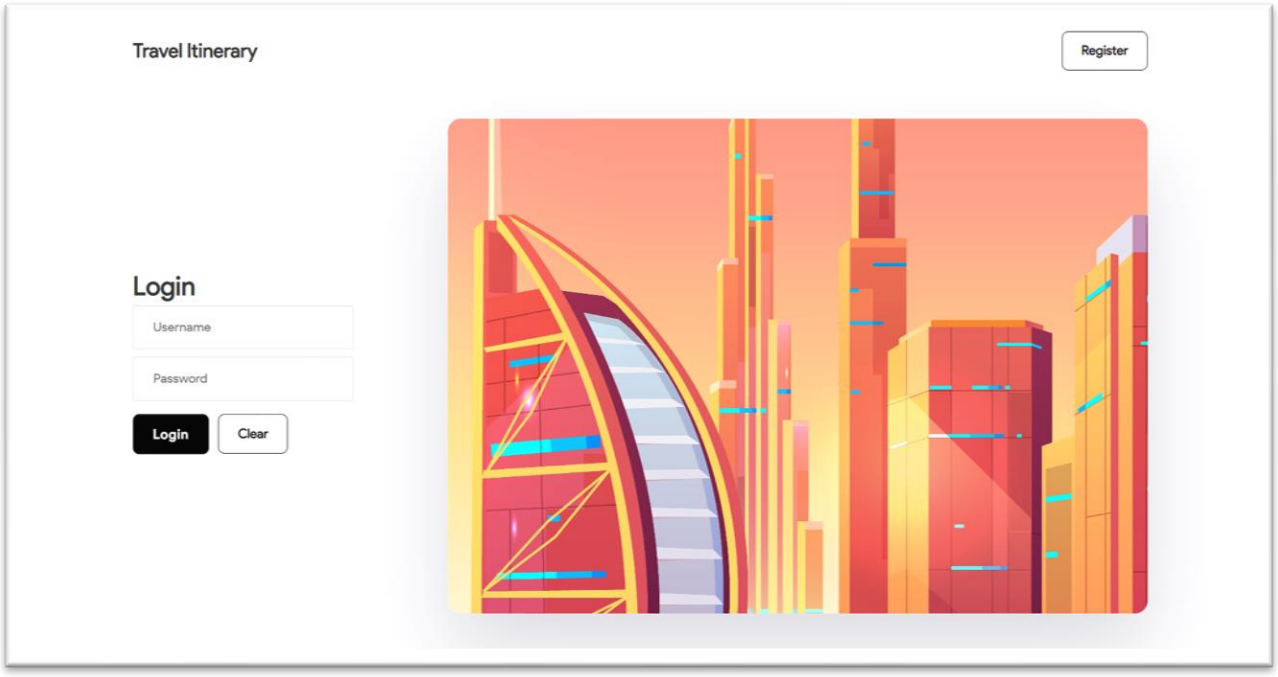

Fig 3: Login Page

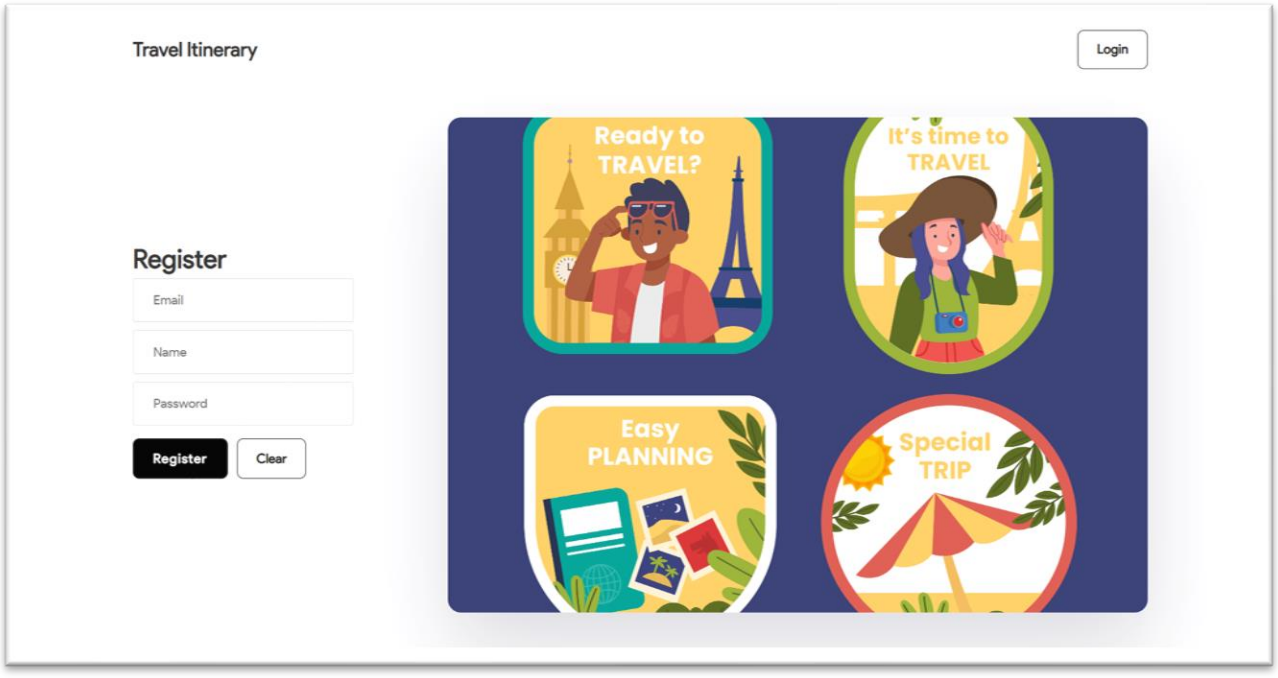

Fig 4: Register Page

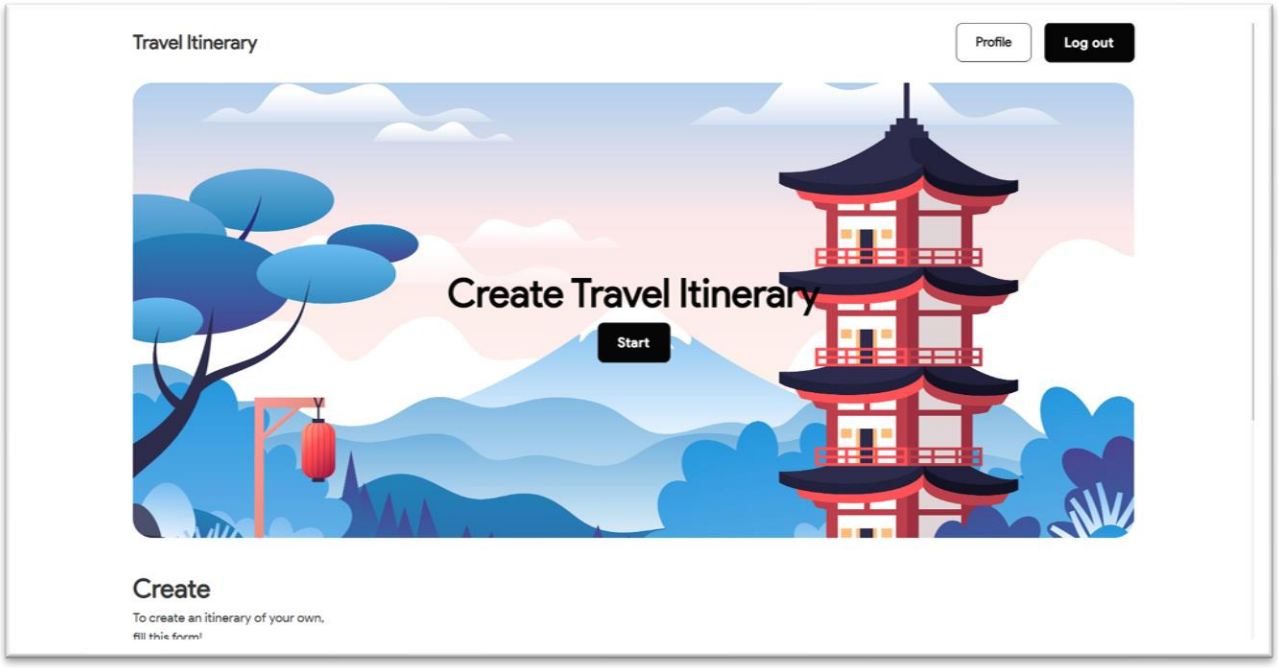

Fig 5: Home Page

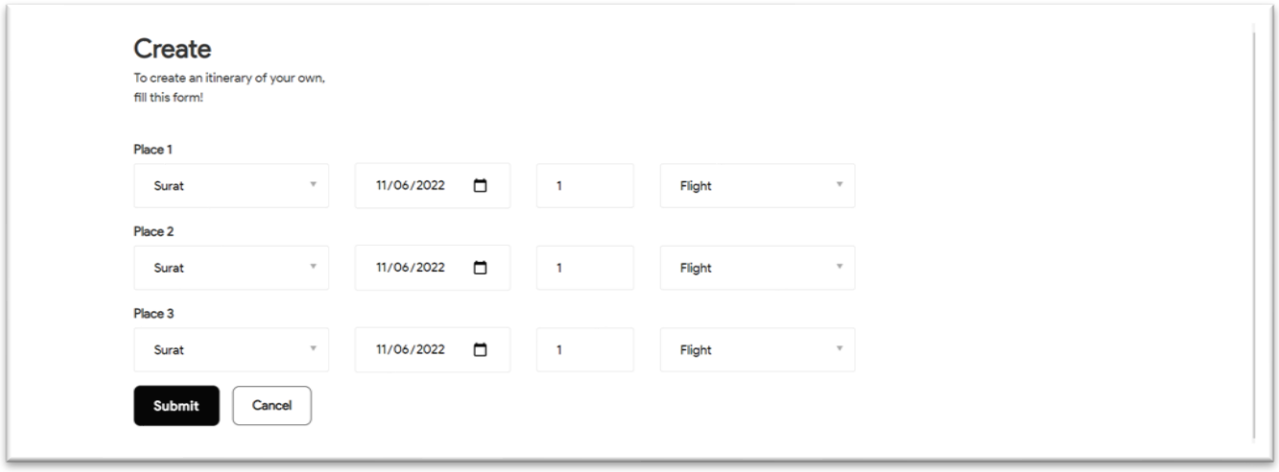

Fig 6: Itinerary form

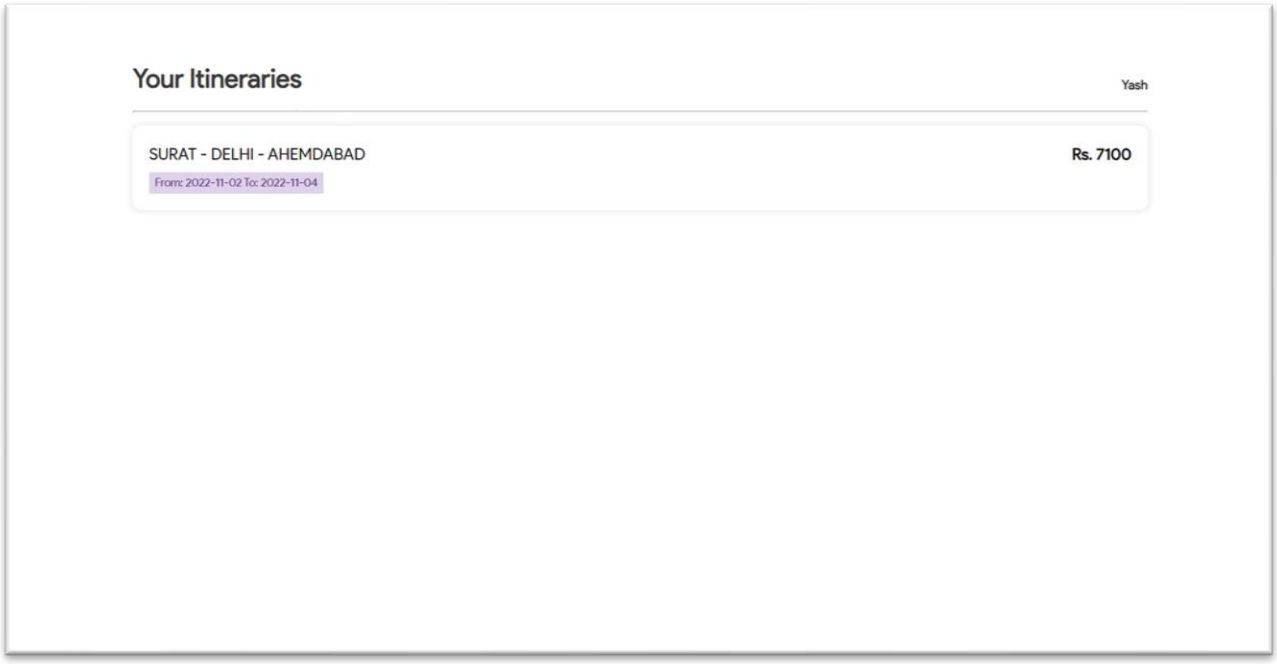

Fig 7: Profile Page

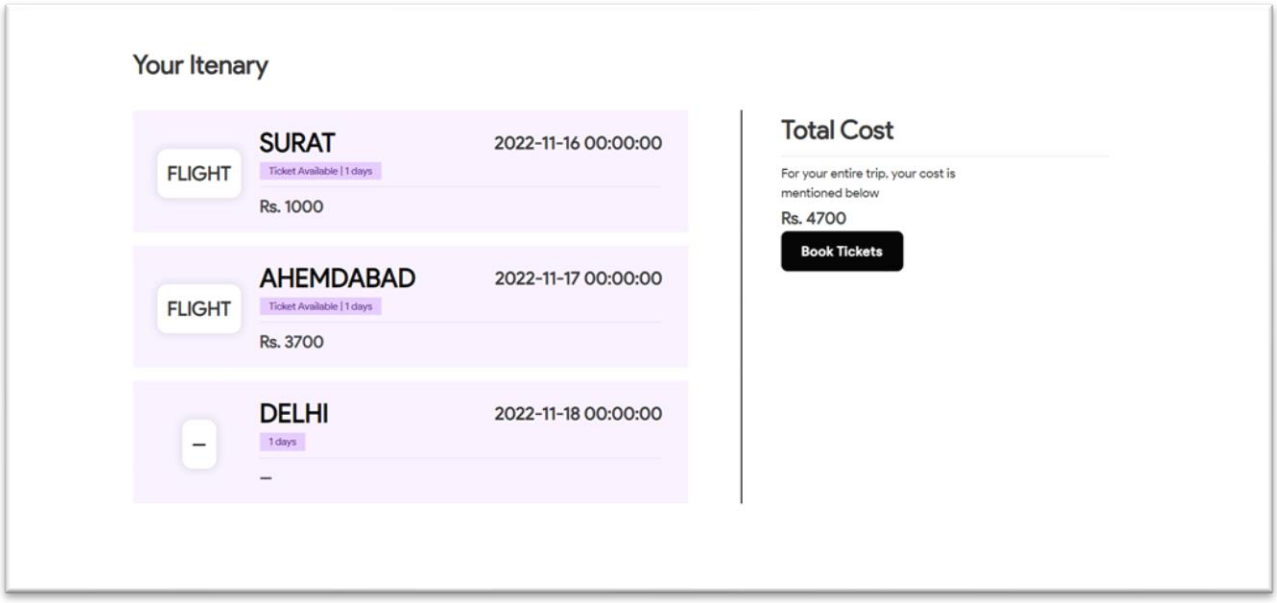

Fig 8: Itinerary Plan and Cost

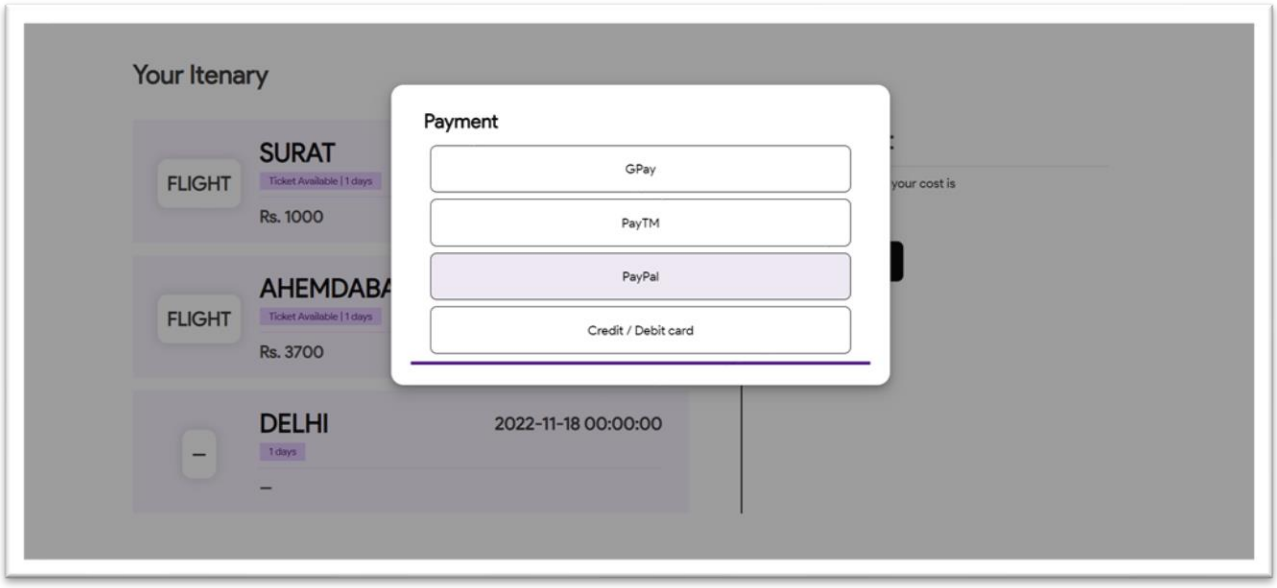

Fig 9: Payment options