Project Journal for Journey Planner System

Participants: Pankaj and Rushikesh
Professor: Prijesh
Project Timeline: 28th September to 8th December 2024

Phase 1: Project Setup and Initialization (28th September - 5th October)

28th September 2024:

-
  - **Team Formation:** Pankaj and Rushikesh decided to collaborate on the project.
  - **Initial Brainstorming:** After discussing multiple ideas, we decided to build a "Journey Planner" system. This system would assist users in finding optimal transport routes based on time, cost, or comfort.

30th September 2024:

- **Requirement Identification:**
  - Discussed the target audience and features with Professor Prijesh.
  - Key features identified:
    1. User registration and login functionality.
    2. Support for both admin and regular users.
    3. Route planning with transport options sorted by user preferences (time, cost, comfort).
    4. Data persistence for user information.
  - Professor Prijesh suggested implementing a basic prototypeof class and OOPS, aloong with its importance and used by mid-October.
    Before his suggestion I was using function only to create this.

5th October 2024:

- **Technology Stack Selection:**
  - Programming Language: C++(as suggested)
  - File I/O for data storage (avoiding complex database setups for simplicity).

Phase 2: User Management System Implementation (6th October - 20th October)

6th October 2024:

- Created the basic structure for the `User` class, including attributes like `name`, `email`, and `password`.
- Introduced `AdminUser` as a derived class of `User`.

10th October 2024:

- Added methods for user sign-up and login functionality.
- Implemented input validation for email and password to ensure proper user credentials.

15th October 2024:

- Enhanced login functionality to differentiate between admin and regular users.
- Tested user registration and login for various scenarios, such as invalid credentials and duplicate emails.

20th October 2024:

- Implemented file-based storage to persist user data (`project.txt`).
- Tested saving and loading user data from the file system.

Phase 3: Route and Transport Option System (21st October - 10th November)

21st October 2024:

- Designed the `Route` and `TransportOption` classes.
- Added attributes to `Transport Option` like `mode`, `time`, `cost`, and `comfort`.

25th October 2024:

-

- Created a basic structure for storing predefined routes and transport options.
- Initialized the route data for a few locations manually.

1st November 2024:

- Added methods to display available routes and transport options.
- Wrote a basic function to sort transport options based on a user's priority (time, cost, or comfort).

10th November 2024:

- Completed the `listRoutes()` function to display all available locations to the user.
- Finalized the sorting mechanism for transport options and integrated user input to determine their preferred priority.

Phase 4: Journey Planning and Payment System (11th November - 25th November)

11th November 2024:

- Implemented the core `planJourney()` function.
- Added logic to accept starting and destination points and validate user input.

15th November 2024:

- Enhanced journey planning to handle scenarios where no direct route exists.
- Improved error messages for better user experience.

20th November 2024:

- Developed the payment processing functionality.
- Ensured seamless integration between transport selection and payment confirmation.

25th November 2024:

- Added user feedback prompts for successful payment and journey planning.

Phase 5: Testing, Debugging, and Enhancements (26th November - 8th December)

26th November 2024:

- Conducted initial testing of all features, including user management, route planning, and transport option sorting.
- Fixed bugs related to input validation and file handling.

30th November 2024:

- Optimized route initialization by reducing repetitive code.
- Improved the display format for transport options to enhance readability.

5th December 2024:

- Final testing phase with mock user data.
- Ensured smooth handling of edge cases, such as invalid location input and unexpected user choices.

8th December 2024:

- Prepared the final presentation for Professor Prijesh.
- Highlighted features, challenges, and lessons learned during the project.

Key Challenges Faced

1.

User Data Persistence:

2.

- Initially, we faced difficulties in implementing file-based storage without overwriting data. This was resolved by appending new data to the file while ensuring duplicates were not created.

3.

Input Validation:

4.

- Handling invalid input gracefully required multiple iterations. For instance, ensuring that an empty email or password does not crash the system.

5.

Route Initialization:

6.

- Manually defining routes was time-consuming, but we prioritized simplicity to avoid unnecessary complexity.

Lessons Learned

1.

Team Collaboration:

2.

- Effective communication between Pankaj and Rushikesh ensured smooth progress. Tasks were divided based on individual strengths.

3.

Incremental Development:

4.

- Building and testing each feature incrementally helped us identify bugs early and avoid major setbacks.

5.

Adaptability:

6.

- We adjusted our approach based on feedback from Professor Prijesh and challenges encountered during implementation.

Notes: Tools like chatGPT is so helpful in optimizating the code and finding the error in the declaration of classes and function.

Submitted by:
Pankaj and Rushikesh

Thanks