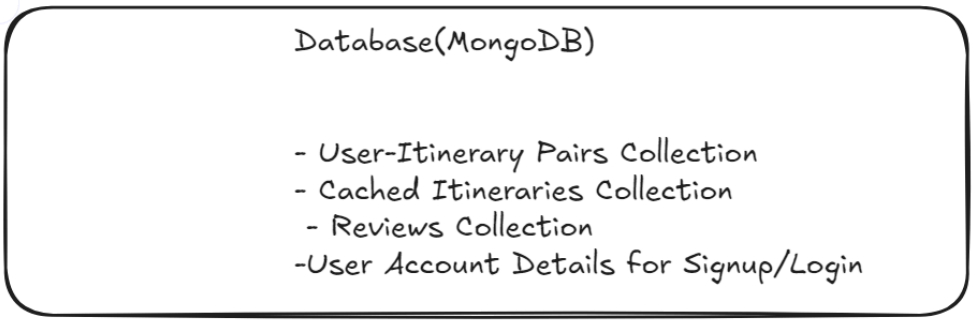
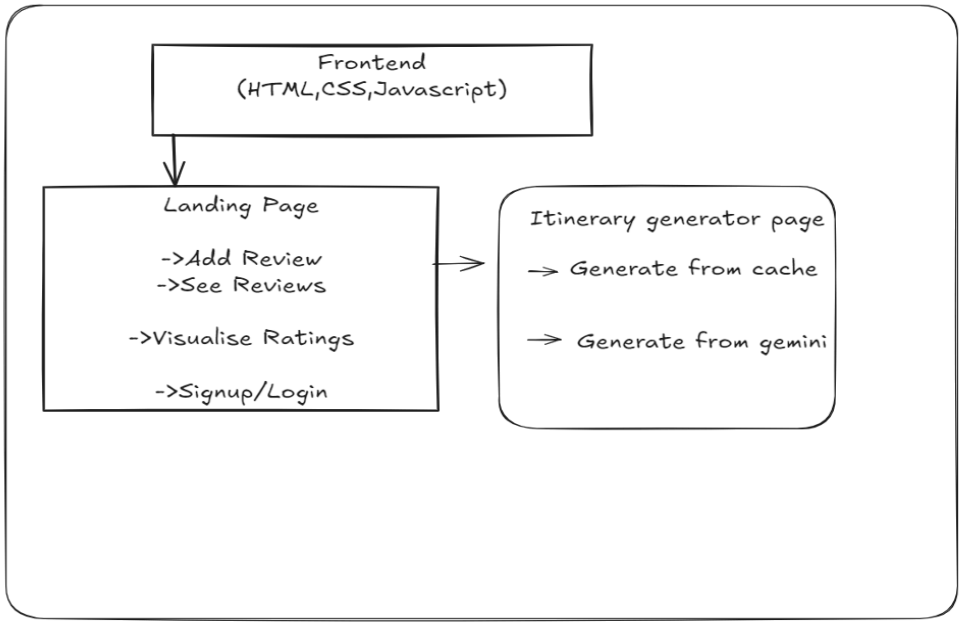
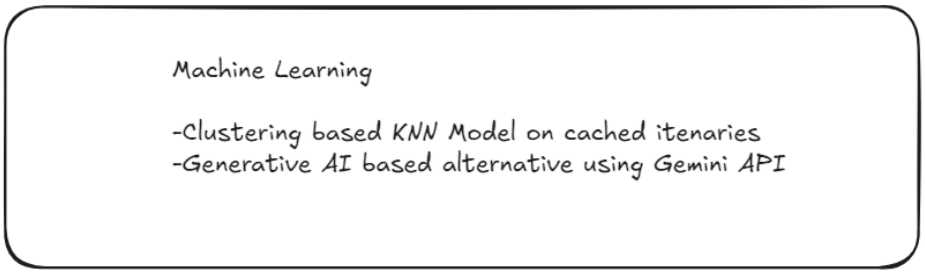
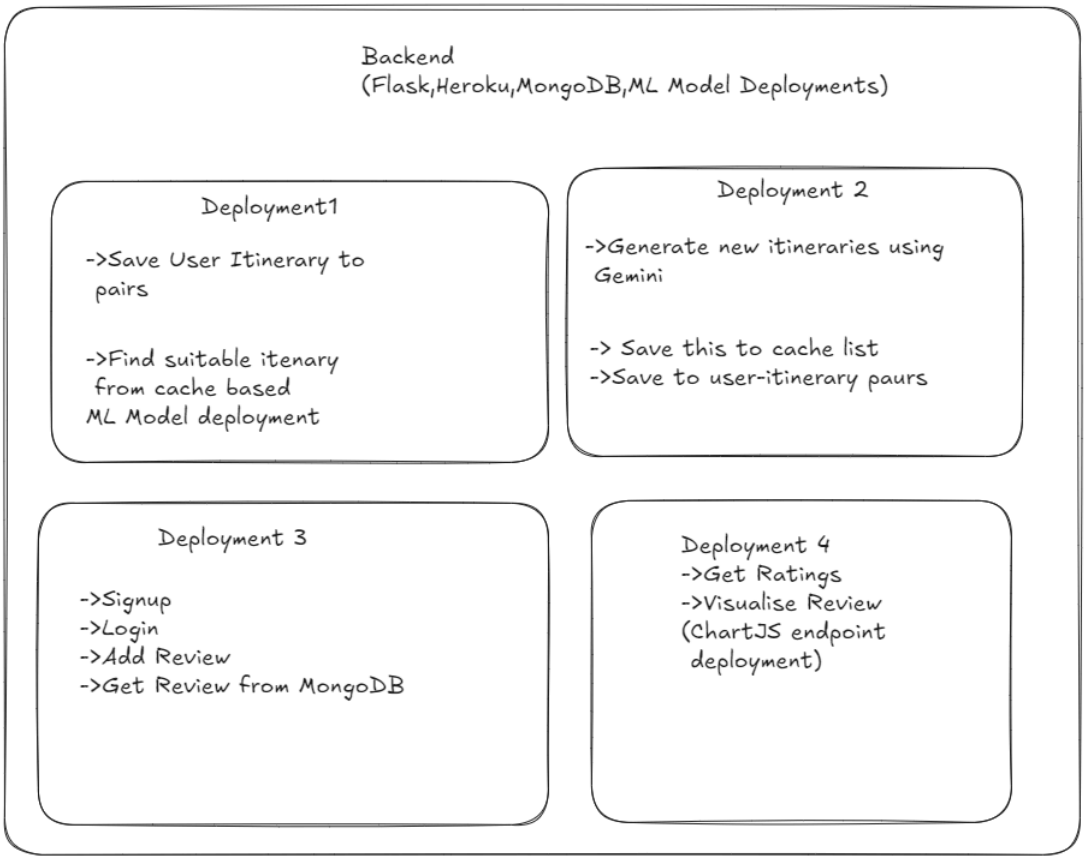
1.Overall Architecture of Application

The following diagrams explain in detail the overall architecture of the application









2.Technology Stack

The personalised itinerary project is built on a wide tech stack that integrates a machine learning model, generative AI, and a MongoDB .

- **Frontend:** The user interface is built with HTML, CSS, and JavaScript. Users interact directly with forms to input their preferences and review itineraries.
- **Backend:** The backend consists of multiple Flask APIs deployed on Heroku, handling tasks such as generating itineraries, managing user authentication, and collecting user feedback.(More details on these APIs in txt file in backend folder of repo)
- **Visualisation**:ChartsJS has been used to dynamically visualise the rating distribution from users directly from the MongoDB database.
- **Machine Learning Model:** A two level ML model : clustering followed by K-Nearest Neighbors (KNN) model trained on a baseline cache of itineraries to recommend the most suitable one based on user inputs.
- **Generative AI:** Powered by the Gemini API, this generates personalised itineraries when the KNN model's recommendation does not fully satisfy the user.
- **MongoDB:** Serves as the database, storing user data, itineraries, reviews, and mappings between users and their selected itineraries.
- **Deployment:** The application is deployed using Heroku for backend APIs and Vercel for frontend hosting.

3.Nuances

1. **Pre-Cached Itineraries for Enhanced Scalability:**Use of pre-cached itineraries to manage computational load and ensure fast response times.
2. **K-Nearest Neighbors (KNN) Model Integration:**Application of the KNN model to recommend the most suitable pre-cached itinerary based on user preferences.
3. **Dynamic Personalization with Generative AI:** Integration of the Gemini API to generate customised itineraries in real-time when pre-cached options are insufficient.
4. **Internal Mapping of User-Itinerary Pairs :**Tracking of which user created or selected specific itineraries to refine recommendations and learn from user interactions.
5. **MongoDB Database Design and Relationships:**Structured database design with collections for itineraries, reviews, user signups, and user-Itinerary pairs, and how they interrelate.
6. **Visualisation of User Ratings with Charts.js:**Use of Charts.js to present user ratings and reviews visually for better understanding and decision-making.

7. Fetching Top Reviews: Implemented API endpoints to directly fetch top reviews from database and render them on frontend.

4.Challenges Faced

(i) Scalability and Performance:

- Solution: Used pre-cached responses to reduce the time needed for generating itineraries. The KNN model quickly finds the best matches based on user preferences, ensuring the app runs smoothly.

(ii) Real-Time Personalization:

- Solution: Added the Gemini API for real-time itinerary creation. If users are unhappy with the recommended itinerary, they can click a "Not OK" button to get a more personalized option instantly.

(iii)Tracking User-Triggered Itineraries:

- Solution: Implemented an internal system to track when users create new itineraries, which can help improve the recommendation system over time.

5.Future Improvement Scope

- **Improving the number of options for personalisation :** With Scale we may attempt to add more interests, say go from four to ten, we may be able to add more complex trips such as not restricting to one particular city to visit and crafting itinerary from 2 to 30 days.
- **Redis for Caching:**Given the tourist statistics by city and monitoring the choices in the database collection itineraries we may cache frequently re-used options for reducing latency and optimising scalability.
- **Automated Re-Training of ML Model:**Automating Re training of ML Model could improve insights that would require lesser use of GenAI tokens since any response every generated by Gemini on the Webapp is directly being stored to itineraries collection along with pre-existing itineraries.