॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड

Indian Institute of Technology Dharwad

PlanSmart

Introduction to PlanSmart

PlanSmart is an innovative event planning platform designed to simplify and enhance the process of organizing events. It offers two distinct approaches to event planning. Users can either take complete control, customizing every aspect of their event from start to finish, or they can utilize the intelligent recommendation system. This system leverages advanced algorithms to suggest the best options for venues, food, decorations, and more, based on user preferences and requirements. With PlanSmart, planning a successful event becomes efficient, tailored, and stress-free.

Author:
Shreya Roy CS24MT019,
Sayali Shirish Malushte CS24MT017,
Subhradip Palit CS24MT009

Date: November 15, 2024

Contents

Site Map

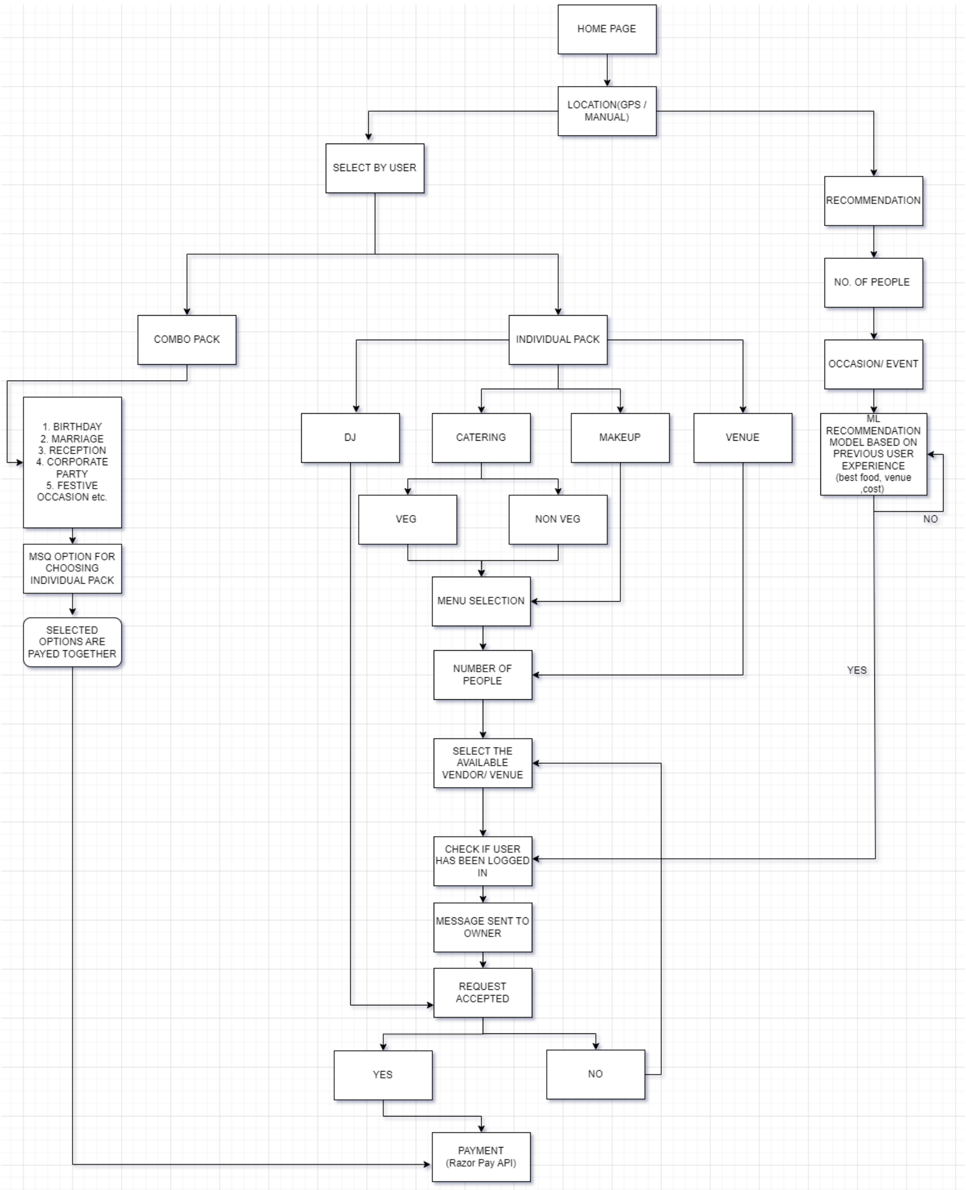

Figure 1.1: Site Map

Dataset

Dynamic Datasets for Vendor and Service Ratings

PlanSmart will utilize datasets sourced from Kaggle to build a comprehensive list of service providers, including:

- **Restaurants**
- **Caterers**
- **Makeup Artists**
- **DJ Services, etc**

Each entry in the dataset will feature vendor details such as name, contact information, service descriptions, and ratings. The platform will ensure that the data remains dynamic and up-to-date through the following mechanisms:

1. **User-Driven Ratings and Reviews**:
   - Users can rate and review service providers after utilizing their services.
   - The platform will dynamically update the ratings, reflecting user feedback to help future users make informed decisions.
   - Consistent feedback patterns may adjust the vendor's ranking and visibility on the platform.
2. **Vendor Profile Management**:
   - Vendors who register on PlanSmart can manage and update their profiles, including service information, availability, pricing, and special offers.
   - Any changes made by the vendor will be reflected in real-time on the platform, ensuring the information remains current and accurate.

By integrating with Kaggle datasets and enabling real-time updates via user feedback and vendor management, PlanSmart guarantees a dynamic, accurate, and reliable platform for event planning.

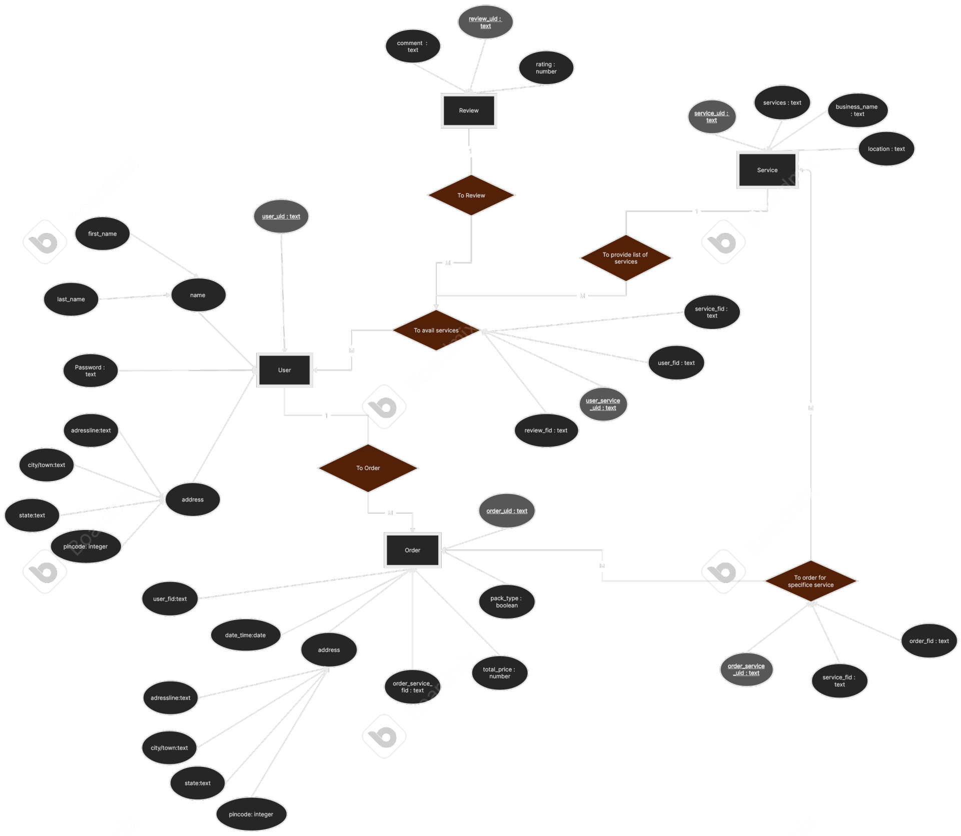

Figure 2.1: ER Diagram

Description Components

3.1 Main Page

- **Selection by User**: If user wants to select services manually according to choice. Here user can select services, respective vendors for services on its own.
- **Location**: GPS shows current location or user has to enter location manually to avail services at that location.

3.2 Select by User

- **Combo Pack**: Many Services clubbed together for a particular occasion. e.g. For Birthday Party user need Catering services, decoration, etc. so this services are selected together to avail offers and to make payment together.
- **Individual Pack**: To avail single service.

3.3 Individual Pack

For different services such as DJ, Venue, Catering, Makeup, etc further process will vary. After Vendor/ Venue is selected and user can proceed to payment, thus order is confirmed.

APIs Used in the PlanSmart Website

In the PlanSmart website, we have integrated the following ready-made APIs to enhance the user experience and streamline the event planning process:

- **Stripe Test Mode:** This payment gateway API is used to facilitate secure transactions. Users can make payments for bookings directly on the platform, ensuring a smooth and reliable payment process.
- **Geolocation API:** This API helps in determining the current location of the user and finding the nearest vendors around them. It allows users to easily discover services and facilities close to their location.

API's Used

- `Post :` `/user/create` : TO create the user in the database
- `Post :` `/user/login` : TO authenticate the user from the database
- `Get :` `/user/<user_id>` : TO get the details of the particular user
- `Post :` `/services/<user_id>` : TO add the service details of the particular user

- `Get : /services/<service_name>` : TO get the details of the service with the particular service_name
- `/services/all` : TO get all the details of all the services
- `/services/unique` : TO get the details of the unique service category names
- `/services/service_by_id/<user_id>` : TO get the details of services of the particular user
- `/order/<user_id>/place` : TO create the order for the particular user
- `/order/<user_id>/payment` : TO send the payment details of the particular user for the particular order to the database
- `/order/<user_id>` : TO get the details of the particular user

Tech Stack

- **Database:** Database used for this project is `SQLite`, code for which is written in `SQLAlchemy`.
- **APIs:** APIs are written in `Flask`.
- **Frontend:** `ReactJs` is used to create the Frontend on top of HTML, CSS, Bootstrap, and JavaScript.
- **API Calling:** `Fetch` method of JavaScript is used to call the APIs.
- **Routing:** `React-Router-DOM` is used for routing.
- **Database Viewer:** `DB Browser for SQLite` is used for visualizing the database and manipulating it directly.
- **Initial API Testing:** `Swagger` is used for testing the APIs in the initial phase.
- **Payment Gateway:** `Stripe` is used as the Payment Gateway.
- **CORS Handling:** `flask-cors` is used to handle CORS (Cross-Origin Resource Sharing).

Component Wise Integration(API used in each component and it's description)

- **AddServiceForm Component** API for creating the service for particular user is used here
- **EditAddress Component** Geological API for getting the current location
- **Footer Component** API for submitting a mail to our email to submit any query or to contact us
- **Individual Component** API for getting unique service categories from the database
- **Invoice Component** API for placing the order of the particular user
- **Login Component** API for logging in the user
- **MyOrders Component** API for getting the orders of a particular user
- **MyServices Component** API for getting the services of the particular user
- **Navbar Component** API for getting unique service categories from the database
- **Payment Component** API for posting the payment details to our database to the stripe
- **Signup Component** API for creating the user in the database
- **Vendors Component** API for getting the services by service category