* **API DISTRIBUTION:**

1. **Tanay**
   1. [Provider Mandap APIs](#_5._Mandap_APIs)
   2. User APIs
      1. [Dashboard APIs](#_1._Dashboard_APIs)
2. **Akshay**
   1. [**Provider APIs**](#_Mandap_Provider_APIs)
      1. Register
      2. Login
      3. Get Provider Details
      4. Get All Bookings for Particular Provider
   2. **User**
      * 1. [Order APIs](#_2._Order_APIs)
3. **Vaishnavi**
   1. **User APIs**
      1. [Mandap APIs](#_3._Mandap_APIs)
   2. **Admin**
      1. [User APIs](#_3._User_APIs)
4. **Kunal**
   1. [**User APIs**](#_User_APIs)
      1. Register
      2. Login
      3. Get User Details
   2. **Admin**
      1. [Provider APIs](#_2._Provider_APIs)
      2. [Socket io implementation of Provider.](#_6._Socket.IO_Implementation)

* **GitHub Repository Setup**
  + **Akshay –** Admin Client, User Client

**Schemas**

Provider model, Mandap Book Model

* + **Tanay –** Provider Client

**Schemas**

Mandap Model, photographer Model, caterer Model

* + **Vaishnavi –** Admin Server, Project Assets

**Schemas:**

Reviews Model, Room Model

* + **Kunal –** User Provider Server

**Schemas:**

Address Model, User Model

**Mandap Provider APIs**

**1. Register**

* **POST /api/provider/register**
  + **Purpose**: Register a new mandap provider.
  + **Fields**: Name, Email, Phone, Password, isApproved (default: false).

**2. Login**

* **POST /api/provider/login**
  + **Purpose**: Authenticate a mandap provider.
  + **Fields**: Email, Password.

**3. Get Provider Details**

* **GET /api/provider/profile**
  + **Purpose**: Retrieve the logged-in provider's details.

**4. Get All Bookings for Particular Provider**

* **GET /api/provider/bookings**
  + **Purpose**: Fetch all bookings associated with the provider, including calendar, bookings, and payment details.

**5. Mandap APIs**

* **POST /api/provider/mandap**
  + **Purpose**: Add a new mandap.
  + **Fields**:
    - Mandap Name
    - Available Dates (array of dates)
    - Venue Types (enum: Banquet hall, Lawn, Resort, Hotel, Community Hall, Farmhouse, Convention Centre, Rooftop Venues)
    - Location (State, City, Pin-code, Full Address)
    - Penalty Charges per Hour
    - Cancellation Policy (enum: No Refund, Partial Refund, Full Refund)
    - Venue Images/Video (multiple, uploaded via Cloudinary)
    - Guest Capacity
    - Venue Pricing
    - Security Deposit
    - Security Deposit Type (enum: Refundable, Non-Refundable)
    - Caterer (Name, Food Type [Veg, Non-Veg, Jain], Plan [Basic, Standard, Premium, Luxury], Menu Items [array of {menuItem, price}], Plan Price)
    - Photographer (Name, Photography Type [Candid, Traditional, Cinematic, Drone, Pre-wedding, Portrait, Album Photography], Price for Type, Sample Work Images [via Cloudinary])
    - Rooms & Accommodation (AC Rooms Count, Non-AC Rooms Count, AC Room Price/Day, Non-AC Room Price/Day)
    - Amenities (array of amenities)
    - Indoor Facilities (Power Backup [Generator, Inverter, Both, None], WIFI [Available, Not Available], Elevator [Available, Not Available])
    - Payment Details (Payment Methods Accepted [UPI, Credit/Debit Card, Cash, Net Banking, Bank Transfer])
* **PUT /api/provider/mandap/:mandapId**
  + **Purpose**: Update an existing mandap by ID.
  + **Fields**: Same as Add Mandap, with optional updates to specific fields.
* **DELETE /api/provider/mandap/:mandapId**
  + **Purpose**: Delete a mandap by ID.
* **GET /api/mandap/search**
  + **Purpose**: Search mandaps by name, city, or guest capacity.
  + **Query Parameters**: name, city, capacity.
* **GET /api/mandap/:mandapId/reviews**
  + **Purpose**: Retrieve all reviews for a specific mandap.

**6. Socket.IO Implementation**

* **Event: booking.created**
  + **Purpose**: Notify the provider in real-time when a new booking is created for their mandap.
* **Event: booking.updated**
  + **Purpose**: Notify the provider when a booking is updated.
* **Event: booking.cancelled**
  + **Purpose**: Notify the provider when a booking is cancelled.
* **Event: review.added**
  + **Purpose**: Notify the provider when a new review is added to their mandap.

**User APIs**

**1. User APIs**

* **POST /api/user/register**
  + **Purpose**: Register a new user.
  + **Fields**: Full Name, Email, Phone Number, Password, Address.
* **POST /api/user/login**
  + **Purpose**: Authenticate a user.
  + **Fields**: Email, Password.
* **GET /api/user/profile**
  + **Purpose**: Retrieve the logged-in user's details.

**2. Order APIs**

* **GET /api/user/bookings**
  + **Purpose**: Fetch all bookings made by the user.
* **POST /api/user/bookings**
  + **Purpose**: Create a new booking.
  + **Fields**:
    - Choose Date (array of strings)
    - Choose Photographer (optional: Photographer ID, Photography Type)
    - Choose Caterer (optional: Caterer ID, Plan, Menu Items)
    - Choose Rooms (optional: AC Rooms Count, Non-AC Rooms Count).
* **PUT /api/user/bookings/:bookingId**
  + **Purpose**: Update an existing booking by ID.
  + **Fields**: Same as Add Booking, with optional updates.
* **DELETE /api/user/bookings/:bookingId**
  + **Purpose**: Cancel/delete a booking by ID.
* **GET /api/user/bookings/:bookingId**
  + **Purpose**: Retrieve details of a specific booking (optional).
* **POST /api/user/payments/razorpay**
  + **Purpose**: Initiate a payment via Razorpay for a booking.
  + **Fields**: Booking ID, Amount.
* **POST /api/user/payments/razorpay/verify**
  + **Purpose**: Verify Razorpay payment success.
  + **Fields**: Razorpay Payment ID, Order ID, Signature.

**3. Mandap APIs**

* **GET /api/mandaps**
  + **Purpose**: Retrieve details of all mandaps.
* **GET /api/mandaps/search**
  + **Purpose**: Search mandaps by city, mandap name, venue types, or available dates.
  + **Query Parameters**: city, mandapName, venueTypes, availableDates.
* **GET /api/mandaps/filter**
  + **Purpose**: Filter mandaps by price or guest capacity.
  + **Query Parameters**: minPrice, maxPrice, minCapacity, maxCapacity.
* **GET /api/mandaps/:mandapId**
  + **Purpose**: Get details of a specific mandap by ID.
* **GET /api/mandaps/featured**
  + **Purpose**: Retrieve featured mandaps (optional).
* **GET /api/mandaps/:mandapId/availability**
  + **Purpose**: Check availability of a specific mandap for given dates.
  + **Query Parameters**: dates (array of dates).
* **POST /api/user/favourites**
  + **Purpose**: Add a mandap to the user’s favourites.
  + **Fields**: Mandap ID.
* **GET /api/user/favourites**
  + **Purpose**: Retrieve all favourite mandaps of the user.
* **DELETE /api/user/favourites/:mandapId**
  + **Purpose**: Remove a mandap from the user’s favourites.
* **POST /api/mandaps/:mandapId/reviews**
  + **Purpose**: Add a review for a mandap.
  + **Fields**: User ID, Provider ID, Mandap ID, Rating, Message.
* **PUT /api/mandaps/:mandapId/reviews/:reviewId**
  + **Purpose**: Update an existing review by ID.
  + **Fields**: Rating, Message (optional updates).
* **DELETE /api/mandaps/:mandapId/reviews/:reviewId**
  + **Purpose**: Delete a review by ID.
* **Get /api/mandaps/:mandapId/reviews/:reviewId**
  + **Purpose**: Get an existing review by ID.

**4. Cloudinary Integration**

* **POST /api/upload/cloudinary**
  + **Purpose**: Upload images/videos (e.g., venue images, photographer sample work) to Cloudinary.
  + **Fields**: File (image/video), Type (venue, photographer).

**5. Socket.IO Implementation (Provider)**

* **Event: booking.confirmed**
  + **Purpose**: Notify the user in real-time when a booking is confirmed.
* **Event: booking.updated**
  + **Purpose**: Notify the user when their booking is updated.
* **Event: booking.cancelled**
  + **Purpose**: Notify the user when their booking is cancelled.
* **Event: mandap.availability.updated**
  + **Purpose**: Notify the user when the availability of a favourite mandap changes.

**Admin APIs**

**1. Dashboard APIs**

* **POST /api/admin/login**
  + **Purpose**: Authenticate an admin.
  + **Fields**: Email, Password.
* **GET /api/admin/users**
  + **Purpose**: Retrieve details of all users.
* **GET /api/admin/providers**
  + **Purpose**: Retrieve details of all providers.

**2. Provider APIs**

* **PUT /api/admin/providers/:providerId/approve**
  + **Purpose**: Approve or reject a provider’s registration request.
  + **Fields**: isApproved (boolean).
* **GET /api/admin/providers/search**
  + **Purpose**: Search providers by name, email, or other criteria.
  + **Query Parameters**: name, email.

**3. User APIs**

* **GET /api/admin/users/search**
  + **Purpose**: Search users by name, email, or other criteria.
  + **Query Parameters**: name, email.