Stoopi Documentation

Introduction:

Technologies Used:

- Runtime Environment: Node.js (express)
- Database: Mongodb
- View Engine: HBS(handlebars),Blade Templates
- Pattern : MVC (Model,View,Controller)

Project Location:
The project is hosted on the server at the location:
Path: /var/www/html/Stoopi-Web
website Path : /var/www/html
Project start: pm2 start Stoopi.js
Project stop: pm2 stop Stoopi.js

Introduction about web

Dashboard:

Introduction about admin-panel

Folder Structure:

Here's an outline of the project's folder structure:

Controllers Folder:

The **controllers** folder contains all the backend mobile API code. Each controller within this folder handles specific API endpoints and related functionality. Controllers are responsible for processing requests and sending responses back to the client.

Public:

In the "Public" folder, you'll find static assets such as CSS, JavaScript, and images that are accessible to clients.
Path: /var/www/html/Stoopi-web/public

View:

The "View" folder is where Handlebars templates reside, providing dynamic content rendering.

Routes.js:

The routes.js file serves as a central routing configuration for the entire project. In this file, you will find the definitions for all the routes associated with each controller. Routes define how incoming requests are mapped to specific controller functions, allowing for proper request handling and execution of the corresponding API logic.
Path: /var/www/html/Stoopi-web/routes.js

Stoopi.js:
The index.js file serves as the entry point for the Node.js server. It plays a crucial role in initializing and starting the server. This file contains the necessary code to set up the server, define middleware, and listen for incoming HTTP requests.
Path: /var/www/html/Stoopi-web/Stoopi.js

1. admin Folder:
The admin folder is dedicated to housing all the Blade view files related to the admin web panel of your application. These view files are responsible for rendering the various pages and components that make up the admin dashboard, including but not limited to user management, analytics, and settings.

Inside the admin folder, you'll find Blade view files organized according to their respective functionality or page. This separation helps maintain a structured and modular approach to your admin panel's frontend code.

2. common Folder:
The common folder contains Blade view files that are commonly shared across multiple parts of your admin panel. Specifically, it typically includes the header and footer components that are consistent throughout the admin interface. These common elements provide a consistent user experience and design across different admin panel pages.
By centralizing these elements in the common folder, you can easily update and maintain the header and footer without needing to make changes in multiple places within your admin panel views

Admin Dashboard:
The dashboard showcases statistics including total active meals, total users, total orders placed, total meal posted and map which conatins all user's last active location.
In this we are getting longitude and latitude to get users location in map.

Controller defination:

Dashboard Controller that extends the Controller class. It has an index method that retrieves the total number of users, orders, and meals from their respective models. It also retrieves the locations of users who have non-null last_lat and last_long values. If the timezone is set in the session, it retrieves the current time and date in that timezone and retrieves the number of active meals that are not sold out and have a meal_ready_by time that is later than the current time. The method returns a view with the total number of users, orders, meals, active meals, and locations as variables.
Controller Path:

Stoopi API Documentation

Authentication API

Admin Login API

Endpoint:
`POST https://app.stoopi.app/admin/login`

Description:
This endpoint allows admin users to log in to the Stoopi platform.

```
{
  "email": "admin@example.com",
  "password": "your_password"
}
```

Admin Dashboard API

Dashboard Overview

Endpoint:
`POST https://app.stoopi.app/admin/dashboard`

Description:
This endpoint retrieves key statistics and data for the admin dashboard.

2. User Management API

Endpoint:

Post `https://app.stoopi.app/admin/users`

Description:

Retrieves a list of registered users with pagination and filtering options.

Authentication:

- Required: **Yes**

- Type: **Bearer Token (JWT)**

3. User Delete API

Endpoint:

Post `https://app.stoopi.app/admin/user-delete/{user_id}`

Description:

Deletes a specific user from the system using their unique user ID. This action is irreversible and should only be performed by authorized admin users.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

4. User View API

Endpoint:

Post `https://app.stoopi.app/admin/user-view/{user_id}`

Description:

Retrieves detailed information about a specific user by their unique user ID. This endpoint is accessible only to authorized admin users.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

5. User Update Status API

Endpoint:

Post `https://app.stoopi.app/admin/user-update-status/{user_id}`

Description:

Updates the status of a specific user. This can be used to activate, deactivate, suspend, or change other status-related attributes of a user. Only authorized admin users can perform this action.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

6. User Verify Profile API

Endpoint:

Post `https://app.stoopi.app/admin/user-verifyprofile/{user_id}`

Description:

Verifies the profile of a specific user. This can be used to update the user's verification status (e.g., verified, pending, rejected). Only authorized admin users can perform this action.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

7. Upload API

Endpoint:

POST `https://app.stoopi.app/admin/upload`

Description:

Allows admins to upload images along with details such as title, description, category, condition, and location.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

Key: image | Value: [upload multiple image files]
Key: title | Value: "Modern Art Painting"
Key: description | Value: "A beautiful abstract painting."
Key: category | Value: "Art & Collectibles"
Key: condition | Value: "new"
Key: location | Value: "New York, USA"

8. Active Posts API

Endpoint:

POST `https://app.stoopi.app/admin/active-posts`

Description:

Retrieves a list of active posts created by users. This endpoint allows admins to view and manage active user-generated posts.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

9. Category List API

Endpoint:

POST `https://app.stoopi.app/admin/category`

Description:

Retrieves a list of categories added by the admin. This endpoint allows admins to manage and view available categories.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

10. Delete Category API

Endpoint:

POST `https://app.stoopi.app/admin/category-delete`

Description:

Deletes an existing category added by the admin. Only authorized admins can perform this action.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

11. Edit Category API

Endpoint:

POST `https://app.stoopi.app/admin/category-edit`

Description:

Updates an existing category. Admins can change the category name using this API.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

12. Get Bad Words List API

Endpoint:

POST `https://app.stoopi.app/admin/bad-words`

Description:

Retrieves a list of restricted/banned words that users are not allowed to use on the platform.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

13. Add Bad Words API

Endpoint:

POST `https://app.stoopi.app/admin/addbad-words`

Description:

Allows an admin to add new restricted/banned words to the list. These words will be filtered from user-generated content.

Authentication:

- Required: **Yes**
- Type: **Bearer Token (JWT)**

14. Get Reports List API

Endpoint:

POST `https://app.stoopi.app/admin/reports`

Description:

Retrieves a list of reports submitted by users regarding inappropriate content or behavior.

Authentication:

- **Required: Yes**
- **Type: Bearer Token (JWT)**

15. View Report Details API

Endpoint:

POST `https://app.stoopi.app/admin/reportsview/{report_id}`

Description:

Allows an admin to view details of a specific report.

Authentication:

- **Required: Yes**
- **Type: Bearer Token (JWT)**

16. Delete Report API

Endpoint:

POST `https://app.stoopi.app/admin/reportsdelete/{report_id}`

Description:

Allows an admin to delete a report after reviewing it.

Authentication:

- **Required: Yes**
- **Type: Bearer Token (JWT)**

Stoopi API Documentation Application

Authentication API App side

12. Crypto Get Keys API

Endpoint:

POST `https://app.stoopi.app/api/v1/crypto/get-keys`

Description:

Retrieves cryptographic keys for secure transactions or authentication purposes. Only authorized users can access this endpoint.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

13. Crypto Encrypt API

Endpoint:

POST `https://app.stoopi.app/api/v1/crypto/encrypt`

Description:

Encrypts the provided data using cryptographic methods. The request body should contain the data to be encrypted.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{
  "data": "Sensitive information"
}
```

14. Crypto Decrypt API

Endpoint:

POST `https://app.stoopi.app/api/v1/crypto/decrypt`

Description:

Decrypts the provided encrypted data and returns the original plain text.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{
  "data": "Sensitive information"
}
```

15. User Registration API

Endpoint:

POST `https://app.stoopi.app/api/v1/auth/register`

Description:

Registers a new user with the provided details. This API requires basic user information, location data, and device details.

Authentication:

- **Required:** No (Public API)

Request Headers:

No headers required.

```
{
  "fullName": "John Doe",
  "email": "john@example.com",
  "username": "john_doe",
  "password": "securepassword123",
  "latitude": 40.7128,
  "longitude": -74.0060,
  "postalCode": "10001",
  "city": "New York",
  "country": "USA",
  "deviceToken": "xyz123abc",
  "deviceId": "device-001",
  "deviceType": "iOS"
}
```

16. User Verification API

Endpoint:

POST `https://app.stoopi.app/api/v1/verifyUser`

Description:

Verifies a user's email using a One-Time Password (OTP) sent to their registered email address.

Authentication:

- **Required:** No (Public API)

Request Headers:

No headers required.

```
{
  "email": "john@example.com",
  "OTP": "123456"
}
```

17. Resend OTP API

Endpoint:

POST `https://app.stoopi.app/api/v1/re-sendotp`

Description:

Resends a One-Time Password (OTP) to the user's registered email for verification purposes.

Authentication:

- **Required:** No (Public API)

Request Headers:

No headers required.

```
{
  "email": "john@example.com"
}
```

18. User Login API

Endpoint:

POST `https://app.stoopi.app/api/v1/auth/login`

Description:

Allows a registered user to log in by providing valid credentials.

Authentication:

- **Required:** No (Public API)

Request Headers:

No headers required.

```
{
  "email": "john@example.com",
  "password": "SecurePassword123",
  "deviceId": "abc12345",
  "deviceType": "Android"
}
```

19. Reset Password API

Endpoint:

POST `https://app.stoopi.app/api/v1/auth/reset-password`

Description:

Allows users to request a password reset by providing their registered email. If the email exists, a reset link or OTP will be sent.

Authentication:

- **Required:** No (Public API)

Request Headers:

No headers required.

```
{
  "email": "john@example.com"
}
```

20. User Logout API

Endpoint:

POST `https://app.stoopi.app/api/v1/auth/logout`

Description:

Logs out a user and invalidates their session by removing the associated device ID.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{
  "deviceId": "abc12345"
}
```

21. Get User Profile API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/profile`

Description:

Retrieves the profile details of the currently authenticated user.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

22. Update User Bio API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/update-bio`

Description:

Updates the bio of the authenticated user.

Authentication:

- **Required:** Yes
-

```
{
  "bio": "Passionate traveler and food lover."
}
```

23. Update Profile Image API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/update-profile-image`

Description:

Updates the profile image of the authenticated user.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{
  "image": "base64_encoded_image_or_image_url"
}
```

24. Get All User Posts API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/post/all`

Description:

Retrieves all posts created by the authenticated user.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

25. Get Active User Posts API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/post/active`

Description:

Retrieves all active posts created by the authenticated user. Only posts with a status of `"AVAILABLE"` will be returned.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

26. Get Favorite User Posts API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/post/favorite`

Description:

Retrieves a list of posts that the authenticated user has marked as favorite.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

27. Remove Post from Favorites API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/post/remove-favorite`

Description:

Removes a post from the authenticated user's list of favorite posts by updating its `isFavorite` status to `false`.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{
  "postId": "98765",
  "isFavorite": "false"
}
```

28. Delete Post API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/post/delete`

Description:

Deletes a post created by the authenticated user. Only the post owner can perform this action.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{
  "postId": "654321"
}
```

29. Get Followers API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/followers`

Description:

Retrieves a list of users who are following the authenticated user.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

ChatGPT said:

31. Get Followers API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/followers`

Description:

Retrieves a list of users who are following the authenticated user.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

33. Update User Location API

Endpoint:

POST `https://app.stoopi.app/api/v1/update-user-location`

Description:

Updates the user's location details, including longitude, latitude, postal code, city, and country.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

```
{

  "longitude": 78.9629,

  "latitude": 20.5937,

  "postalCode": "110001",
  "city": "New Delhi",

  "country": "India"

}
```

34. Social Login API

Endpoint:

POST `https://app.stoopi.app/api/v1/socialLogin`

Description:

Allows users to log in or register using social authentication (e.g., Google, Facebook, Apple). If the user is new, an account is created; otherwise, they are logged in.

Authentication:

- **Required:** No (handled via social authentication provider)

```
{

  "fullName": "John Doe",

  "email": "johndoe@example.com",

  "social_id": "1234567890",

  "social_type": "google",

  "deviceId": "abcdef12345",

  "deviceType": "iOS",

  "deviceToken": "xyzpushnotificationtoken",

  "username": "john_d",

  "register": true,

  "latitude": 37.7749,

  "longitude": -122.4194

}
```

35. Leaderboard - Top Creators API

Endpoint:

POST `https://app.stoopi.app/api/v1/leaderboard/top-creators`

Description:

Fetches a list of top content creators based on their post count. Users who are deleted or blocked are excluded from the leaderboard.

Authentication:

Filter Conditions:

The API applies the following filters internally:

- Excludes deleted users (`deleteUser: null`)
- Excludes blocked users (`isBlocked: { $ne: true }`)
- Includes only users with at least one post (`postcount: { $gte: 1 }`)

36. Create Post API

Endpoint:

POST `https://app.stoopi.app/api/v1/post/create`

Description:

Allows users to create a new post with details such as title, description, location, and category.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

Field	Type	Required	Description
title	String	Yes	Title of the post

description	String	Yes	Description of the post
categoryId	String	Yes	ID of the category
condition	String	Yes	Condition of the item
longitude	Number	Yes	Longitude of the post location
latitude	Number	Yes	Latitude of the post location
postalCode	String	No	Postal code of the location
street	String	No	Street name
city	String	Yes	City name
address	String	No	Full address
country	String	Yes	Country name
bringAFriend	Boolean	No	Indicates if a friend can join

37. Update Post API

Endpoint:

POST `https://app.stoopi.app/api/v1/post/update`

Description:

Allows users to update an existing post by modifying details such as title, description, category, location, and condition.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

Request Body:

Field	Type	Required	Description
postId	String	Yes	The unique ID of the post to update
title	String	No	Updated title of the post
description	String	No	Updated description of the post
categoryId	String	No	Updated category ID
condition	String	No	Updated item condition
longitude	Number	No	Updated longitude
latitude	Number	No	Updated latitude
postalCode	String	No	Updated postal code
street	String	No	Updated street name
city	String	No	Updated city name
address	String	No	Updated full address

country	String	No	Updated country name
bringAFriend	Boolean	No	Update whether a friend can join

38 Update Account API

Endpoint:

POST `https://app.stoopi.app/api/v1/account/update`

Description:

This API allows users to update their account details, including email, full name, and username.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

39 Delete Account API

Endpoint:

POST `https://app.stoopi.app/api/v1/account/delete`

Description:

This API allows a user to permanently delete their account. Once deleted, the account cannot be recovered.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

40. Report a Post API

Endpoint:

POST `https://app.stoopi.app/api/v1/report/post`

Description:

Allows users to report a post for violating community guidelines.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

41. Find Posts API

Endpoint:

POST `https://app.stoopi.app/api/v1/post/find`

Description:

Searches for posts based on user-provided filters.

Authentication:

- **Required:** No

42. Find Posts Near Me API

Endpoint:

POST `https://app.stoopi.app/api/v1/post/nearme`

Description:

Fetches posts that are near the user’s current location.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)

43. Get Post Categories API

Endpoint:

POST `https://app.stoopi.app/api/v1/post/categories`

Description:

Retrieves a list of available post categories.

Authentication:

- **Required:** No

---

44. Get User Feed API

Endpoint:

POST `https://app.stoopi.app/api/v1/own/feed`

Description:

Retrieves a personalized feed of posts based on user preferences and interactions.

Authentication:

- **Required:** Yes
- **Type:** Bearer Token (JWT)