

User API Documentation

1. Register User

****Endpoint:**** `POST /api/register`

Registers a new user with the provided details.

****Request Body:****

- `UserName` (string): The username for the new user. Must be 3-20 characters long and should not contain spaces.
- `Name` (string): The full name of the user. Must be 3-50 characters long.
- `Password` (string): The password for the new user. Must be at least 8 characters long, contain at least one lowercase letter, one uppercase letter, one number, and one special character.
- `CPassword` (string): Confirmation of the password. Must match the `Password`.
- `PhoneNo` (string): The phone number of the user. Must be a valid Indian phone number.
- `EmailID` (string): The email address of the user. Must be a valid email.

****Responses:****

- `201 Created`: Registration successful.
- `400 Bad Request`: Validation errors, such as invalid email, short name, mismatched passwords, etc.
- `500 Internal Server Error`: Unexpected server errors.

2. Login User

****Endpoint:**** `POST /api/login`

Logs in a user with the provided credentials.

****Request Body:****

- `UserName` (string, optional): The username of the user. Required if `EmailID` is not provided.
- `EmailID` (string, optional): The email address of the user. Required if `UserName` is not provided.
- `Password` (string): The password of the user. Must be a valid strong password.

****Responses:****

- `200 OK`: Login successful, sets an authentication token.
- `400 Bad Request`: Validation errors, such as invalid credentials.

3. Authenticate User

****Endpoint:**** `GET /api/authenticate`

Authenticates the currently logged-in user.

****Responses:****

- `200 OK`: User is authenticated, returns user details.
- `404 Not Found`: User not found.
- `500 Internal Server Error`: Unexpected server errors.

4. Create Gigs

****Endpoint:**** `POST /api/createGigs`

Creates a new gig with the provided details.

****Request Body:**** Multipart/form-data

- `data` (string): A JSON string containing the gig details:
 - `title` (string): The title of the gig. Must be between 10 and 100 characters long.
 - `cost` (number): The cost of the gig. Must be at least 100.
 - `description` (string): The description of the gig. Must be between 500 and 1500 characters long.
 - `tags` (array of strings): The tags associated with the gig. Each tag must be between 1 and 50 characters long. A maximum of 10 tags are allowed.
 - `deliveryTime` (number): The delivery time for the gig in hours. Must be at least 1 hour.
 - `category` (string): The category of the gig. Must be between 1 and 50 characters long.
 - `maxPendingOrders` (number): The maximum number of pending orders for the gig. Must be at least 1.
- `files` (array of files): The images associated with the gig. At least 1 and at most 5 images are allowed.

****Responses:****

- `201 Created`: Gig created successfully.
- `400 Bad Request`: Validation errors, such as invalid title, description, category, etc.
- `500 Internal Server Error`: Unexpected server errors.

5. Fetch Gigs

****Endpoint:**** `GET /api/fetchGigs`

Fetches gigs based on the provided query parameters.

****Query Parameters:****

- `title` (string, optional): The title of the gig to search for. Maximum length is 100 characters.
- `tags` (string, optional): Comma-separated tags to search for. Each tag must be a maximum of 50 characters long.

- `category` (string, optional): The category of the gig to search for. Maximum length is 50 characters.
- `recommended` (string, optional): Whether to fetch recommended gigs. Must be either 'true' or 'false'.
- `by` (string, optional): The username of the user who created the gig. Maximum length is 50 characters.
- `gigId` (string, optional): The ID of the gig to fetch. Maximum length is 35 characters.

****Responses:****

- `200 OK`: Gigs fetched successfully.
- `400 Bad Request`: Validation errors, such as invalid title, tags, category, etc.
- `500 Internal Server Error`: Unexpected server errors.

6. Get Media

****Endpoint:**** `GET /api/getmedia/:gid/:image`

Fetches the media file for a specific gig.

****Path Parameters:****

- `gid` (string): The ID of the gig.
- `image` (string): The name of the image file.

****Responses:****

- `200 OK`: Media file fetched successfully.
- `404 Not Found`: File not found.
- `500 Internal Server Error`: Unexpected server errors.

7. Create Order

****Endpoint:**** `POST /api/createOrder`

Creates a new order for a gig.

****Request Body:****

- `gigId` (string): The ID of the gig to order. Must be a valid string with a maximum length of 35 characters.

****Responses:****

- `201 Created`: Order created successfully.
- `400 Bad Request`: Validation errors, such as invalid gig ID.
- `500 Internal Server Error`: Unexpected server errors.

8. Fetch Orders

****Endpoint:**** `GET /api/fetchOrders`

Fetches orders for the logged-in user based on the provided query parameters.

****Query Parameters:****

- `status` (string, optional): The status of the orders to fetch. Must be one of 'cancelled', 'pending', 'completed', or 'accepted'.
- `orderId` (string, optional): The ID of the order to fetch. Must be a valid string with a maximum length of 35 characters.

****Responses:****

- `200 OK`: Orders fetched successfully.
- `400 Bad Request`: Validation errors, such as invalid status, order ID, etc.
- `500 Internal Server Error`: Unexpected server errors.

9. Get Notifications

****Endpoint:**** `GET /api/getNotification`

Fetches notifications for the logged-in user.

****Responses:****

- `200 OK`: Notifications fetched successfully.
- `500 Internal Server Error`: Unexpected server errors.

10. Create Wallet

****Endpoint:**** `POST /api/createWallet`

Creates a new wallet for the logged-in user.

****Responses:****

- `201 Created`: Wallet created successfully.
- `500 Internal Server Error`: Unexpected server errors.

Will introduce some condition in future for creation of wallet

11. Cancel Order

****Endpoint:**** `POST /api/cancelOrder`

Cancels an existing order.

****Request Body:****

- `orderId` (string): The ID of the order to cancel. Must be a valid string with a maximum length of 35 characters.

****Responses:****

- `200 OK`: Order cancelled and amount refunded successfully.
- `400 Bad Request`: Validation errors, such as invalid order ID.
- `403 Forbidden`: User does not have permission to cancel the order.
- `500 Internal Server Error`: Unexpected server errors.