# User API Documentation

This documentation outlines the available endpoints for managing user accounts, including login, registration, retrieval, update, and deletion.

## User Endpoints

### 1. User Login

* Endpoint: `/api/user/login`  
  Method: `POST`
* Request Body:

{  
 "mPIN": "1234"  
}

* Response: Returns an authorization key and the user ID.

{  
 "Authorization": "Bearer generated\_key",  
 "LoginUserId": "1"  
}

### 2. User Onboarding (Sign Up)

#### Personal Details

* Endpoint: `/api/user/signup/personalDetails`  
  Method: `POST`
* Request Body:

{  
 "firstName": "John",  
 "lastName": "Doe",  
 "mobileNumber": "1234567890",  
 "aadharNumber": "123456789012",  
 "email": "john.doe@example.com",  
 "dateOfBirth": "YYYY-MM-DD",  
 "mPIN": "1234"  
}

* Response: Returns user details upon successful registration.

{  
 "userId": "1",  
 "firstName": "John",  
 "lastName": "Doe",  
 "mobileNumber": "1234567890",  
 "aadharNumber": "123456789012",  
 "email": "john.doe@example.com",  
 "dateOfBirth": "YYYY-MM-DD",  
 "createdAt": "date",  
 "updatedAt": "date"  
}

#### Address Details

* Endpoint: `/api/user/signup/addressDetails`  
  Method: `POST`  
  Request Parameter: `userId` (generated upon registration)
* Example: `userId=1`
* Request Body:

{  
 "street": "123 Main St",  
 "city": "City",  
 "state": "State",  
 "zipcode": "123456"  
}

* Response: "User registered successfully"

### 3. Get User by ID

* Endpoint: `/api/user/getuser/{userId}`  
  Method: `GET`
* Response: Returns the user DTO object if found, or an appropriate message if not.

{  
 "Id": "1",  
 "firstName": "John",  
 "lastName": "Doe",  
 "mobileNumber": "1234567890",  
 "email": "john.doe@example.com",  
 "dob": "YYYY-MM-DD",  
 "role": ["Agent"],  
 "createdAt": "date",  
 "updatedAt": "date"  
}

### 4. Get All Users

* Endpoint: `/api/user/getusers`  
  Method: `GET`
* Response: Returns a list of all users.

[  
 {  
 "Id": "1",  
 "firstName": "John",  
 "lastName": "Doe",  
 "mobileNumber": "1234567890",  
 "email": "john.doe@example.com",  
 "dob": "YYYY-MM-DD",  
 "createdAt": "date",  
 "updatedAt": "date",  
 "isVerified": "verified/not verified"  
 },  
 {  
 "Id": "2",  
 "firstName": "Jane",  
 "lastName": "Doe",  
 "mobileNumber": "0987654321",  
 "email": "jane.doe@example.com",  
 "dob": "YYYY-MM-DD",  
 "createdAt": "date",  
 "updatedAt": "date",  
 "isVerified": "verified/not verified"  
 }  
]

### 5. Update User

* Endpoint: `/api/user/updateuser/{mpin}`  
  Method: `PUT`
* Request Body:

{  
 "mobileNumber": "1234567890",  
 "email": "john.doe@example.com"  
}

* Response: Returns updated user details.

{  
 "Id": "1",  
 "firstName": "John",  
 "lastName": "Doe",  
 "mobileNumber": "1234567890",  
 "email": "john.doe@example.com",  
 "dob": "YYYY-MM-DD",  
 "role": ["Agent"],  
 "createdAt": "date",  
 "updatedAt": "date",  
 "isVerified": "verified/not verified"  
}

### 6. Delete User

* Endpoint: `/api/user/deleteuser`  
  Method: `DELETE`
* Request Body:

{  
 "userId": "1"  
}

* Response: "Successful" or "Failure"

### 7. Change mPIN

* Endpoint: `/api/user/updateMpin`  
  Method: `POST`
* Request Body:

{  
 "mPIN": "1234",  
 "newMpin": "4321"  
}

* Response: "Successful" or "Failure"

## User Address Details Endpoints

User registration and update processes include necessary validations and null checks to ensure data integrity.