**Django URL Mappings Documentation:**

**User Views:**

This document provides an overview of the URL mappings for the user-related views in your Django project. The URLs defined here map the user views to their corresponding endpoints, allowing the application to handle various user-related functionalities such as registration, login, profile management, password reset, and account deletion.

**Import Statements:**

* `**from django.urls import path, include**`: Imports the `**path**` and `**include**` functions from Django’s URL configuration module. These functions are used to define URL patterns and include other URL configurations.
* **`from api.views.user\_views import \*`**: Imports all the views related to user operations from the `**user\_views**` module within the `**api.views**` package.

**URL Patterns for User Views:**

1. **User Registration:**

* **URL:** `/api/user/register/`
* **View**: `RegisterView`
* **HTTP Method**: `POST`
* **Description**: Handles user registration by accepting user details (e.g., name, email, password) and creating a new user account.
* **Request Body** (JSON):

{

"first\_name": "John",

"last\_name": "John",

"user\_name": "John",

"email": "john.doe@example.com",

"password": "SecurePassword123",

"password2": "SecurePassword123"

}

1. **User** **Login:**

* **URL:** `/api/user/ login/`
* **View**: ` LoginView`
* **HTTP Method**: `POST`
* **Description**: Authenticates the user by verifying the email and password. If valid, it returns an authentication token.
* **Request Body** (JSON):

{

"email": "john.doe@example.com",

"password": "SecurePassword123"

}

**3. User Profile:**

* **URL: `**/api/user/profile/`
* **View:** UserView
* **HTTP Method:** GET
* **Description:** Retrieves the profile details of the authenticated user.

**4. User Logout**

* **URL: ‘/**api/user/logout**/’**
* **View:** LogoutView
* **HTTP Method:** POST
* **Description**: Logs out the authenticated user by invalidating their authentication token.
* **Request Body** (JSON):{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

**5. Password Reset Request**

* **URL**: ‘/api/user/reset-password/’
* **View**: ResetPasswordView
* **HTTP Method**: POST
* **Description**: Initiates the password reset process by sending a reset link or code to the user's registered email.
* **Request Body** (JSON):{

"email": "john.doe@example.com"

}

**6. Verify Password Reset**

* **URL**: ‘/api/user/verify-reset-password/’
* **View**: VerifyResetPasswordView
* **HTTP Method**: POST
* **Description**: Verifies the password reset link or code and allows the user to set a new password.
* **Request Body** (JSON):{

"otp": "\*\*\*\*\*\*.",

"new\_password": "NewSecurePassword123",

"confirm\_new\_password": "NewSecurePassword123"

}

**7. Delete User Account**

* **URL**: ‘/api/user/delete-user/’
* **View**: DeleteUserView
* **HTTP Method**: DELETE
* **Description**: Deletes the user account from the system.

**8. Update User Profile**

* **URL**: ‘/api/user/update-user/’
* **View**: UserUpdateView
* **HTTP Method**: PUT/PATCH
* **Description**: Updates the user's profile information such as name, email, etc.
* **Request Body** (JSON):{
* "first\_name": "John",
* "last\_name": "John",
* "user\_name": "John",
* "email": "john.doe@example.com",
* "password": "SecurePassword123",
* "password2": "SecurePassword123"
* }

**9. Top Stories:**

* **URL: `**/api/top-stories/`
* **View:** TopStoriesView
* **HTTP Method:** GET
* **Description:** Retrieves the top trending financial stories.

**10. chat-bot limiter:**

* **URL: `**/api/bot/`
* **View:** ChatbotView
* **HTTP Method:** POST
* **Description:** A chat-bot limiter refers to a set of controls or constraints applied to a chatbot's functionality to ensure it operates within specific guidelines or limitations. This can help manage its performance, security, and user experience.
* **Request Body** (JSON):{

“message”:”मषावबपमबकजापबकजपबकजापबक”

}

* **Request Authorization**: {

**“token”**: “adfsfkjhakldshfo;eifuoiosdapihfsdiaufhypsiaoufayosdfhiasudf”

}