BACKEND DEVELOPER TEST DOCUMENTATION

[REPO - LINK]

1. TEST CASE

I. Login Endpoint (api/login/):-

→ Valid Login:

- 1. Input: Valid username and password.
- 2. Expected Output: Successful login with a status code of 200 and a response message.

→ Invalid Login:

- 1. Input: Invalid username or password.
- 2. Expected Output: Unauthorised access with a status code of 401 and an appropriate error message.

II. File Upload (api/upload/):-

→ Valid File Upload:

- 1. Input: Upload a file with a valid format.
- 2. Expected Output: Successful file upload with a status code of 201 and a success message.

→ Invalid File Type Upload:

- 1. Input: Attempt to upload a file with an invalid format.
- 2. Expected Output: Rejection of the file upload with a status code of 400 and an error message.

III. User Signup (api/signup/)

→ Valid User Signup:

- 1. Input: Provide valid user data.
- 2. Expected Output: Successful user registration with a status code of 201 and a registration confirmation message.

→ Invalid User Signup:

- 1. Input: Attempt to sign up with invalid user data.
- 2. Expected Output: Rejection of user registration with a status code of 400 and appropriate error messages.

IV. Email Verification (api/verify-email/uid/token/):-

→ Valid Email Verification:

- 1. Input: Use a valid verification link.
- 2. Expected Output: Successful email verification with a status code of 200 and a verification confirmation message.

→ Invalid Email Verification:

1. Input: Use an invalid or expired verification link.

- 2. Expected Output: Unsuccessful email verification with a status code of 400 and an error message.
- V. Download file (api/download-file/file-id/) :-

→ Valid File Download:

- 1. Input: Request to download a file with a valid file ID.
- 2. Expected Output: Successful file download with a valid download link and a status code of 200.
- VI. List All Uploaded Files (api/list-uploaded-files/):-

→ List Uploaded Files:

- 1. Input: Request to list all uploaded files.
- 2. Expected Output: Successfully retrieve a list of uploaded files with a status code of 200.

2. DEVELOPMENT PLAN

I. Environment Setup :-

- => Ensuring the production environment is ready by installing all the necessary tools for running Django applications.
- => By Keeping things organised using a requirements.txt file to manage dependencies.

II. Database Migration (PostgreSQL):

- => Making sure the database is set up correctly by running Django's migration commands.
- => Using PostgreSQL for the database as it's reliable and performs well.
- III. Web Server Configuration (Nginx): Configuring Nginx to deal with static files and direct requests to the Django app.

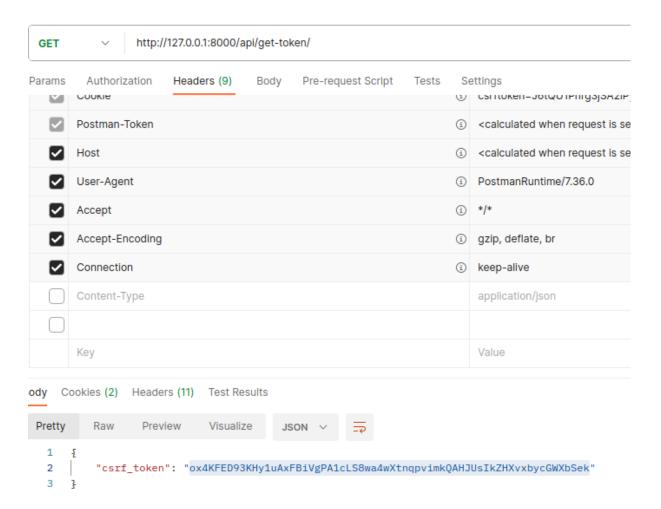
IV. Security Measures:

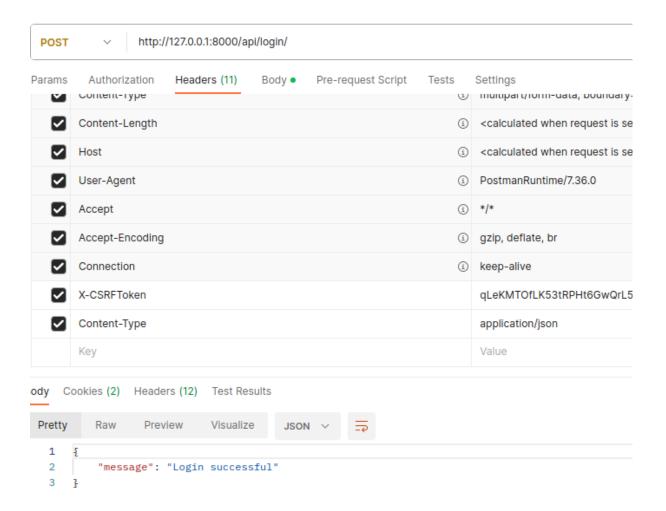
- => Securing data in transit by enabling HTTPS.
- => Keeping the server safe by following security best practices and setting up firewalls.
- **V. Email Configuration:** Setting up email configurations (e.g., SMTP) for user verification and improved communication.
- **VI. Documentation:** Regularly updating and maintaining a clear documentation outlining the backend development process.

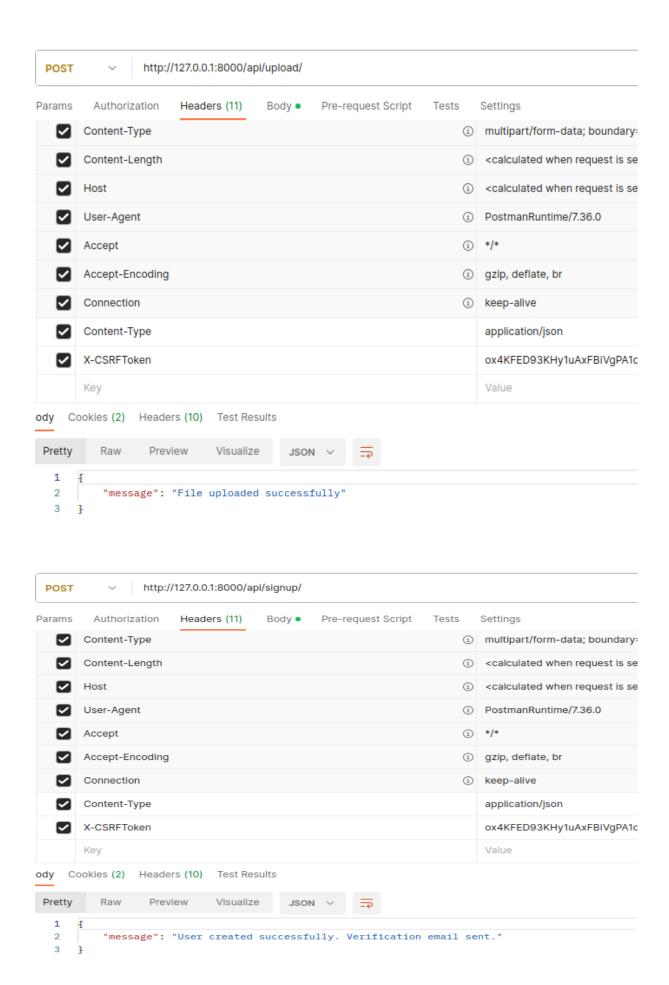
VII. Post Deployment Monitoring:

- => Implement processes for post-deployment monitoring using Render to ensure ongoing stability.
- => Developing effective communication strategies to inform users about new features and updates.

REPONSE SNAPSHOTS







GMAIL VERIFICATION:-

to me ▼

 $Click the following link to verify your email: \underline{http://127.0.0.1:8000/api/verify-email/NQ/c0dzwv-630536b1bf2f8a41dfb8c1a9c14f7a60}$

