PROJECT REQUIREMENT DOCUMENT

TITLE: Register and Login Pages

INTRODUCTION:

The goal of the " Register and Login Pages" project is to provide a foundation or starting point for developers looking to implement user authentication and registration functionality in their web applications. It may serve as a learning resource or a boilerplate codebase that can be customized and extended based on specific project requirements.

FRONT-END BLOCK:

1. Navigate to 'analyst-ai/frontend' and install dependencies. `npm install`

2. Start the app server. \`npm start’

###Runs the app in the development mode.

Open [http://localhost:3000] to view it in the browser.

3.`npm test`

4.`npm run build`

5.`npm run eject`

DATABASE:

1. Database Connection: The frontend and backend of the project connect to the MySQL database using a suitable database connector or ORM (Object-Relational Mapping) library. This allows the application to interact with the database and perform CRUD (Create, Read, Update, Delete) operations.

2.Query Execution: The backend server executes SQL queries to interact with the MySQL database. These queries can include INSERT statements for user registration, SELECT statements for user login and retrieval, UPDATE statements for updating user information, and DELETE statements for removing user accounts.

3.User Authentication: To handle user authentication, the backend likely verifies user credentials against the stored data in the MySQL database. This involves executing SELECT queries to check if the provided username/email and password match an existing user record. The backend might also employ password hashing techniques to securely store and compare passwords.

4.Database Configuration: The project includes a configuration file or environment variables that specify the MySQL database connection details such as the host, port, database name, username, and password. This allows the application to establish a connection with the MySQL database.

5.Data Validation: The backend might implement data validation checks to ensure the integrity and consistency of user data before performing database operations. This helps prevent invalid or malicious data from being stored in the database.

BACK-END BLOCK:

Start your terminal(Command prompt)

1. Create a virtual environment.

`py -m venv `

2. Activate your virtual environment.

`\Scripts\activate`

3. Navigate to `analyst-ai/back`, and install all dependencies from requirements.txt

`pip install -r requirements.txt`

4. Configure the Database. You can configure your database in `analystai/back/back/settings.py`. Configuration for MySQL and sqlite3 is already provided, you just need to comment one while using the other. While using MySQL, remember to set your db credentials within DATABASES.

```
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.mysql',
'NAME': '',
'USER': '',
'PASSWORD': ''
'HOST': 'localhost',
'PORT': '3306',
}
}
```
'''

5. Run migrations.

`py manage.py makemigrations`

`py manage.py migrate`

6. Create admin user or superuser.

`py manage.py createsuperuser --username="" --email="your email"`

Enter your password when prompted.

7. Start the server.

`py manage.py runserver`

TIME-LINE:

The complexity and scale of the " Register and Login Pages" project, you may need to allocate several weeks or even months to complete the development, testing, and deployment phases. It is recommended to create a detailed project plan or work breakdown structure (WBS) to accurately estimate the timeline.

SOURECODE LINK:

https://github.com/vandanapeddireddy/vandana

APPLICATION URL LINK:

http://localhost:3000