Technical Documentation

This technical documentation provides detailed instructions on how to set up and run the project without Docker. The system uses Django for the backend, MongoDB Atlas for data storage, and React.js for the frontend.

1. Setting up the backend (Django)

The project's backend is built with Django, and data is stored using MongoDB Atlas. To set up and run the backend, follow the procedures listed below.

Prerequisites:

- Python 3.x
- MongoDB Atlas account (for database connectivity)

Steps for Running the Backend:

a. Navigate to the backend folder:

Open your terminal and go to the backend folder.

b. Install Virtual Environment:

Command:

- python -m venv venv
- .\venv\Scripts\activate.ps1

c. Install Requirements:

Install all dependencies indicated in the requirements.txt file.

Command:

- pip install -r requirements.txt

d. Configure the MongoDB Atlas:

Update the database connection string in your Django settings file to point to the MongoDB Atlas cluster.

Code:

```
DATABASES = {
   'default': {
       'ENGINE': 'djongo',
       'NAME': '<your-database-name>',
       'CLIENT': {
          'host':             'mongodb+srv://<username>:<password>@cluster0.mongodb.net/<your-database-
name>?retryWrites=true&w=majority'
       }
```

e. Run the Django server:

Start the Django development server by running the following command:

Command:

- python manage.py runserver.

This starts the backend at http://localhost:8000.

2. Setting Up the Frontend (React.js)

The frontend is built using React.js and interacts with the Django backend via REST APIs.

Prerequisites:

- Node.js version 18.x or higher

Steps to Run the Frontend:

a. Navigate to the Frontend Folder:

Open your terminal and navigate to the frontend folder.

b. Install Node.js Packages:

Install the necessary npm packages by running:

Command:

- npm install

c. Run the React Application:

Start the React development server using the following command:

Command

- npm start

This will start the frontend on http://localhost:3000.

3. Connecting the Frontend and Backend

Once both the Django backend and React frontend are running, they will communicate via API requests. The frontend will make HTTP requests to the backend (running on http://localhost:8000 by default) to fetch or send data.

Important API Endpoints:

- User Registration:
- POST /api/register

- User Login:
- POST /api/login

- Time Logging:
- POST /api/timelogs

Ensure that the React application has the correct API endpoint configured for backend interactions.

4. Database Configuration (MongoDB Atlas)

MongoDB Atlas is used for data storage. The connection to MongoDB is managed via the Django djongo package, which allows the use of MongoDB with Django's ORM.

How I did thee Database Setup:

a. Create a MongoDB Atlas Account:

If you don’t have a MongoDB Atlas account, sign up and create a new cluster.

b. Configure the Cluster:

Set up a new database and create a user with read/write access to the database.

c. Update Django Settings:

Add your MongoDB connection string to the Django settings.py file as shown in the earlier section.

5. Production Deployment

Although this setup is for local development, the following steps can be taken for production deployment:

Backend:

You can deploy the Django application to a server like AWS EC2, DigitalOcean, or Heroku. Ensure that you set up the environment variables correctly for production (e.g., using .env files for sensitive information like MongoDB credentials).

Frontend:

The React frontend can be built using npm build and then deployed to a static site hosting service such as Netlify, Vercel, or AWS S3.

Database:

MongoDB Atlas is already hosted in the cloud, so you won't need to make any changes here for production.

6. Notification Service

The **Notification Service** in this project ensures that users are informed of critical events such as missing time logs, project milestones, or task updates. The service is integrated with Django and sends notifications via email or push notifications. This section will describe how to configure, run, and use the notification service in the project.

Notification Service Setup:

To implement and run the notification service, follow the steps below:

1. Email Notifications (SMTP Configuration in Django)

The project uses Django's built-in email backend for sending email notifications. For example, notifications about missing time logs or project completion can be delivered via email to the concerned users.

Prerequisites:

- SMTP server credentials (e.g., Gmail, SendGrid, etc.)