**1. How the Recommendation Works**

The system helps by remembering what you've worked on before and matching it with what you're asking now. It uses stored information to provide relevant responses based on your previous conversations, like ongoing projects or topics you're studying.

**2. How the Matching Algorithm Works**

* Step 1: Understanding the Query: The system breaks down your question into important keywords or topics.
* Step 2: Context Comparison: It checks the stored information (like past projects or interests) to find something related.
* Step 3: Generating the Response: It uses the most relevant match to create an answer, ensuring it’s useful for your current work.

**3. Key Design Choices**

* The system assumes you'll benefit from it remembering key details about your projects or interests.
* It prioritizes recent information because you're likely still working on those topics.
* It adapts responses based on your previous conversations, so they feel personalized.

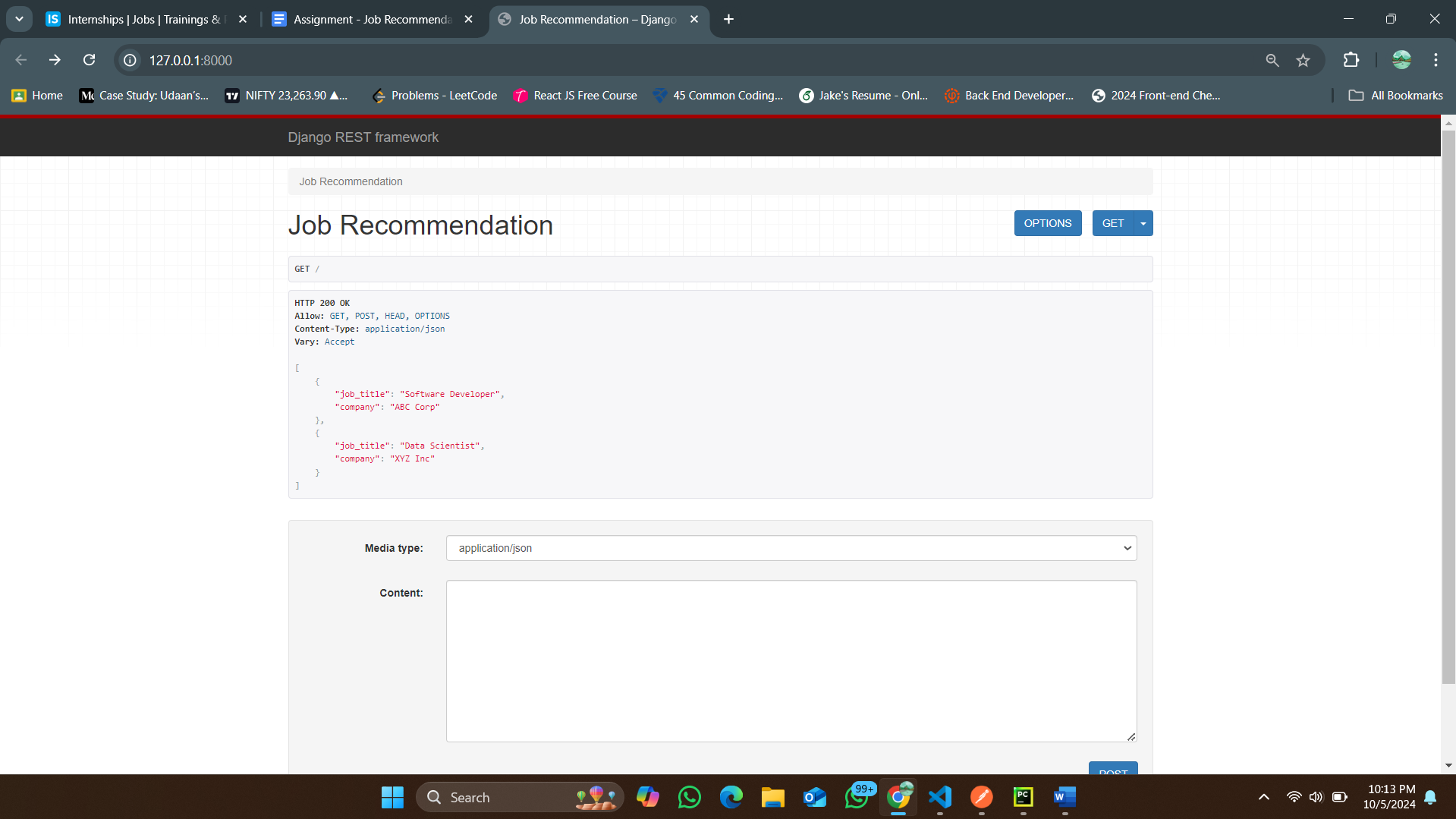
**4. Challenges and Solutions**

* Too Much Context: If there’s too much stored information, the system focuses on the most recent or relevant details.
* Vague Questions: If the question isn’t clear, the system either asks for more details or provides general help.

5. **Important Files and Their Roles**

1. **settings.py**: Contains the configuration for your Django project, including installed apps, middleware, and database settings.
2. **urls.py (project)**: Maps URL patterns to views at the project level.
3. **urls.py (app)**: Maps URL patterns to views at the app level.
4. **models.py**: Defines the database models for user profiles and job postings.
5. **serializers.py**: Contains serializers for converting complex data types like querysets to JSON.
6. **views.py**: Implements the API views to handle incoming requests and return job recommendations.
7. **recommendation\_logic.py**: Contains the logic for matching user profiles with job postings.
8. **requirements.txt**: Lists the necessary packages to install using pip (e.g., Django, Django REST framework).

**Home Page -**



**PostMan –**

