Assignment: Building a Django Backend API for a Path Lab Application

Objective:
Develop a backend API for a path lab application named "Lab Tech" using Django and Django REST Framework (DRF). This project should follow industry standards for backend development, including a structured project organization, robust models, and scalable API endpoints.

Instructions:

1. **Project Setup:**
   - Initialize a Django project named `labtech`.
   - Set up a virtual environment and install the required dependencies (Django, DRF).
   - Configure the project settings for development, including database setup.
2. **Project Structure:**

Organize the project with the following structure:

```
labtech/
├── labtech/        # Project configuration directory
├── app/            # Main application directory
├── templates/      # Templates directory
├── static/         # Static files directory
└── media/          # Media files directory
```

   -
   - Update `manage.py` and `settings.py` to reflect the new project name.
3. **Model Design:**
   - Create models for `Patient`, `Test`, and `LabReport` to store patient details, available tests, and generated lab reports respectively.
   - Use appropriate field types, relationships (e.g., ForeignKey), and constraints.
4. **API Development:**
   - Implement RESTful APIs using DRF for CRUD operations on `Patient`, `Test`, and `LabReport` entities.
   - Create serializers for model data transformation.
   - Use viewsets and a router to structure API endpoints.

5. **URL Configuration:**

    - Configure URLs to include API routes and the Django admin panel.
    - Use versioning for API endpoints (e.g., `/api/v1/`).
6. **Documentation:**

    - Add Swagger or OpenAPI documentation for the API using `drf-yasg` or similar tools.
    - Ensure endpoints are well-documented with descriptions and example requests.
7. **Industrial Practices:**

    - Use database migrations for schema changes.
    - Implement authentication mechanisms for secure API access.
    - Ensure modularity and scalability of the codebase.
    - Follow coding standards and document the code where necessary.
8. **Testing and Deployment:**

    - Test the API using tools like Postman or Swagger UI.
    - Deploy the project on a local server and provide instructions for running it.

Deliverables:

- A fully functional backend API with endpoints for managing patients, tests, and lab reports.
- Swagger or OpenAPI documentation accessible via a web browser.
- A clear README file with setup instructions.

Evaluation Criteria:

- Proper implementation of Django and DRF concepts.
- Code structure, readability, and scalability.
- Completeness of API functionality.
- Quality of documentation and adherence to best practices.

Link to Figma:
https://www.figma.com/design/6FvHZEpinbk9ganfUe8lj0/Labtech?node-id=93-2895&t=tovgJ2W0oVMNIqY3-1