System Requirements

Specification Index

For

Django and React Integration

(Topic:- Django and Front-End Integration)

Version 1.0

Scenario:

You are working as a Django developer in a team that is developing a Restaurant Ordering System. The front-end team has provided a React-based single-page application (SPA) that will handle displaying the menu and submitting orders. Your task is to set up Django as the backend API server, integrate it with the React app, and ensure that the API can handle the operations such as displaying menu items and accepting orders.

Problem Statement

Your task is to:

- **Set up Django as an API server using Django REST framework (DRF).**
- **Implement a MenuItem model to store restaurant menu items in the database.**
- **Implement an Order model to store customer orders.**
- **Create API endpoints to retrieve menu items and accept customer orders.**
- **Configure Django to serve the React front-end and handle CORS (Cross-Origin Resource Sharing).**
- **Ensure that the front-end is integrated with the Django back-end to display the menu and allow order submissions.**

Your Task:

- **Set up Django with Django REST framework to expose an API for managing menu items and orders.**
- **Set up the MenuItem and Order models.**
- **Create the API views for fetching menu items and posting orders.**
- **Make sure the Django project serves the React front-end and provides CORS support.**

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to

   Application menu(Three horizontal lines at left top)->Terminal->NewTerminal.

3. The editor Auto Saves the code.
4. If you want to exit (logout) and to continue the coding later anytime(using Save & Exit option on Assessment LandingPage) then you need to use CTRL+Shift+B command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while

   logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

6. To test any Restful application, the last option on the left panel of IDE, you can find

   ThunderClient, which is the lightweight equivalent of POSTMAN.

7. To test any UI based application the second last option on the left panel of IDE, you can

   find Browser Preview, where you can launch the application.

8. Install ‘djangorestframework’ module before running the code. For this use the following command.

   pip install djangorestframework

9. Use the following command to run the server

   python3 manage.py runserver

10. Mandatory: Before final submission run the following commands to execute testcases

    python3 manage.py test library.test.test_functional

    python3 manage.py test library.test.test_exceptional

    python3 manage.py test library.test.test_boundary

11. To test rest end points

    Click on ‘Thunder Client’ or use Ctrl+Shift+R->Click on ‘New Request’ (at left side of IDE)

12. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.

13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.