# System Requirements Specification Index

**For**

## Managing User Profiles in a Social Networking Platform

**(Topic:-  Django Forms)**

**Version 1.0**

**Problem Statement Description**

**Scenario:**

You have joined a social networking startup as a Django developer. The company is launching its user profile management system, allowing users to personalize their accounts by adding profile pictures, bio, and contact details.

Users should be able to update their profile information conveniently via a Django ModelForm.

**Problem Statement:**

Your task is to design a Django Model Form named UserProfileForm that will be used to manage user profile updates efficiently.

Each user profile should include the following details:

- bio → A short description about the user (CharField, max length: 300).
- profile_picture → A profile image (ImageField, allow only valid image uploads).
- contact_info → User's contact details (CharField, max length: 100).

   **Your Task**

- Define a Django model named UserProfile with the fields mentioned above.
- Create a ModelForm named UserProfileForm to handle form submissions efficiently.
- Ensure form validation for fields like image format and max length constraints.
- Run Django migrations after defining the model.

**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.