



MYSORE UNIVERSITY SCHOOL OF ENGINEERING

Manasagangotri campus, Mysuru-570006
(Approved by AICTE, New Delhi)



UNIVERSITY OF MYSORE

An Assignment(21CD71,Full Stack Development) Report

Submitted By

GUNARI NISITHA SRI

21SECD06

7th Semester,

Department of CS&D

MUSE.

Under Faculty Incharge

- 1) Dr. M. S. Govinde Gowda, Director.
- 2) Mr. Karthik M N
Asst. Professor,
Dept. of CS&D
MUSE.

FULL STACK DEVELOPMENT - 21CD71

SECURE JOB APPLICATION FORM USING DJANGO

Ques: Implement a Secure Job Application Form using Django

- Applicants can submit their name, email, phone number, resume (PDF only), and a cover letter.
- The application form should be implemented using Django Model Forms and should validate:
 - The resume file is in PDF format only.
 - The phone number contains exactly 10 digits.
 - Store all applications in a database and allow the admin to view, shortlist, or reject applications.
- Implement CSRF protection for the form submission.

Prerequisites:

- Operating System: Windows/Linux/MacOS.
- Internet connection.

Steps: 1. Install Python:

- Download the latest version of Python from the official website.
- Run the installer and ensure the option "Add Python to PATH" is checked. This makes python accessible from any terminal window.
- Verify installation by opening a terminal and running:

python --version

```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Manasa MP>python --version
Python 3.12.3

C:\Users\Manasa MP>
```

- We should see the installed Python version.

2. Install Django:

- Open a terminal and install Django using pip (Python's package manager):

>> pip install Django

```
C:\Users\Manasa MP>pip install django
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: django in c:\users\manasa mp\appdata\roaming\python\python312\site-packages (5.1.5)
Requirement already satisfied: asgiref<4,>=3.8.1 in c:\users\manasa mp\appdata\roaming\python\python312\site-packages (from django) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\manasa mp\appdata\roaming\python\python312\site-packages (from django) (0.5.3)
Requirement already satisfied: tzdata in c:\users\manasa mp\appdata\roaming\python\python312\site-packages (from django) (2025.1)

[notice] A new release of pip is available: 25.0 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\Manasa MP>
```

- Verify installation: **python -m django --version**

```
C:\Users\Manasa MP>python -m django --version
5.1.5

C:\Users\Manasa MP>
```

- This ensures Django has been installed successfully.

3. Install Visual Studio Code (VS Code):

- Download VS Code from the official website.
- Install the software by following the on-screen instructions.
- Once installed, add the Python and Django extensions in VS Code for

features like syntax highlighting and debugging support.

4. Verify All Installations:

- Open VS Code, create a new terminal, and check the Python and Django versions using the commands shown above.
- Ensure everything is correctly set up before proceeding.

“SECURE JOB APPLICATION FORM USING DJANGO”

Step 01: Applicants can submit their name, email, phone number, resume (PDF only), and a cover letter.

Name:

Email:

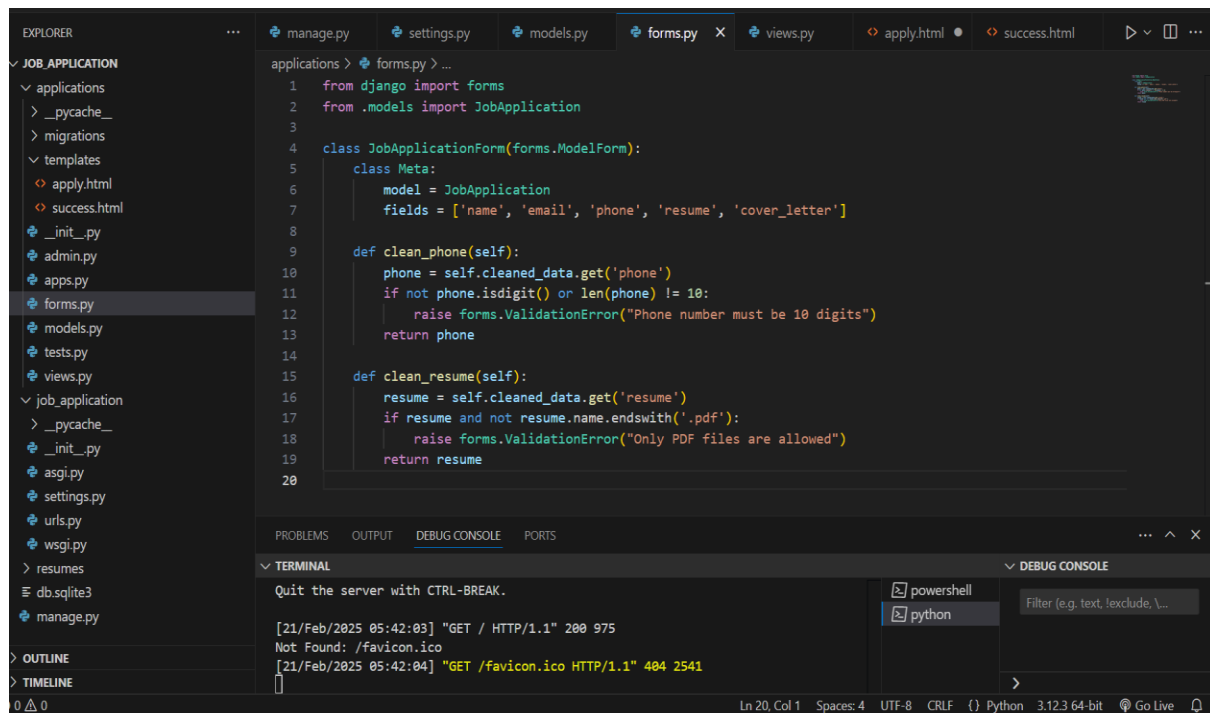
Phone:

Resume: No file chosen

Cover letter:

Step 02:

- The application form should be implemented using Django Model Forms and should validate:
- The resume file is in PDF format only.
- The phone number contains exactly 10 digits.



The screenshot shows a code editor with the following components:

- EXPLORER:** A sidebar on the left showing the project structure. The 'JOB_APPLICATION' folder is expanded, showing files like 'applications', 'templates', 'forms.py', 'models.py', 'views.py', 'urls.py', 'wsgi.py', 'manage.py', 'asgi.py', 'settings.py', 'urls.py', 'wsgi.py', 'db.sqlite3', 'resumes', 'OUTLINE', and 'TIMELINE'.
- forms.py:** The main editor window displays the code for the 'JobApplicationForm' class. The code includes imports for 'django' and 'forms', and 'JobApplication' from 'models'. The class 'JobApplicationForm' inherits from 'forms.ModelForm'. It has a 'Meta' class with 'model = JobApplication' and 'fields = ['name', 'email', 'phone', 'resume', 'cover_letter']'. It also has two methods: 'clean_phone' which checks if the phone number is 10 digits, and 'clean_resume' which checks if the resume file ends with '.pdf'.
- TERMINAL:** A terminal window at the bottom shows the output of the server. It displays the message 'Quit the server with CTRL-BREAK.' and two GET requests: '[21/Feb/2025 05:42:03] "GET / HTTP/1.1" 200 975' and '[21/Feb/2025 05:42:04] "GET /favicon.ico HTTP/1.1" 404 2541'.
- DEBUG CONSOLE:** A panel on the right side of the terminal window, currently empty.

Step 03:

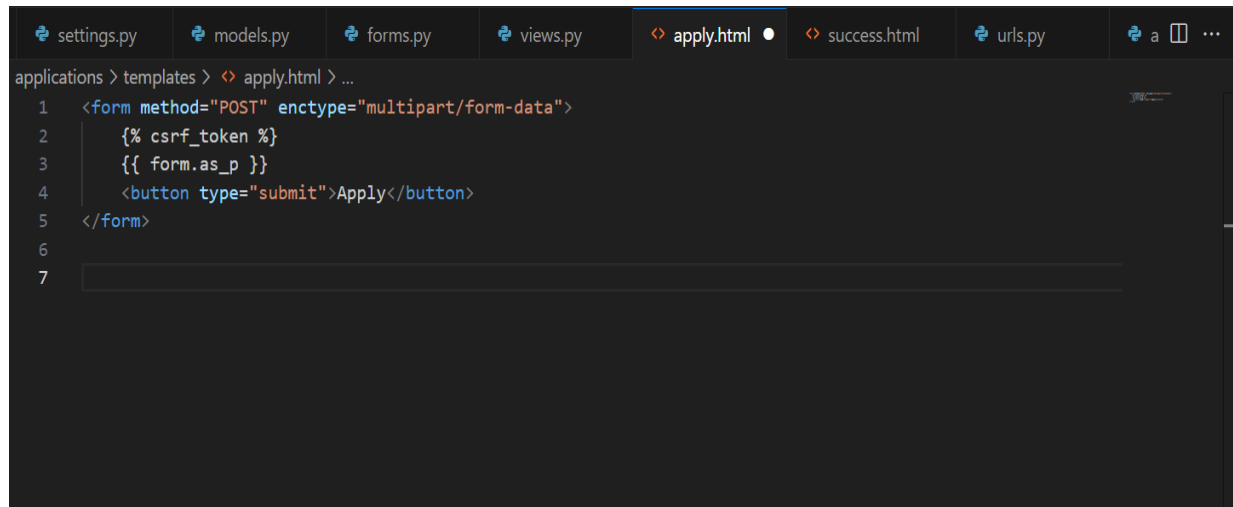
- Store all applications in a database and allow the admin to view, shortlist, or reject applications.

The screenshot shows a web application interface for managing job applications. On the left is a sidebar with a search bar and two main sections: 'APPLICATIONS' (containing 'Job applications' with an '+ Add' button) and 'AUTHENTICATION AND AUTHORIZATION' (containing 'Groups' and 'Users', each with an '+ Add' button). The main content area is titled 'Change job application' and features a 'HISTORY' button in the top right. The form is for an application named 'manasamp'. It includes input fields for 'Name' (manasamp), 'Email' (manasampcd@gmail.com), and 'Phone' (7676782486). The 'Resume' section shows the current file 'resumes/Manasa_Mp_Resume_1_10d9XU.pdf' and a 'Choose File' button. Below this is a 'Cover letter' text area containing the word 'resume'. At the bottom, there is a 'Status' dropdown menu currently set to 'Pending', and a row of buttons: 'SAVE', 'Save and add another application', 'Save and continue editing', and 'Delete'.

This is a close-up view of the 'Status' dropdown menu from the form. The label 'Status:' is visible on the left. The dropdown menu is open, showing three options: 'Pending' (which is highlighted with a dark grey background), 'Shortlisted', and 'Rejected'. Below the dropdown, parts of the 'SAVE', 'Save and add another application', and 'Save and continue editing' buttons are visible.

Step 04:

Implement CSRF protection for the form submission.



The screenshot shows a code editor with several tabs: settings.py, models.py, forms.py, views.py, apply.html (active), success.html, and urls.py. The active tab, apply.html, displays the following HTML code:

```
1 <form method="POST" enctype="multipart/form-data">
2     {% csrf_token %}
3     {{ form.as_p }}
4     <button type="submit">Apply</button>
5 </form>
6
7
```
