INTERNSHIP REPORT

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

by

RAGUL A

23AD050

Under Supervision of

Mr. Athithya

Pinesphere Solutions**., Coimbatore**

(Period: 23/06/2025 to 07/07/2025)

Learn Beyond

KPR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous, NAAC 'A')

Avinashi Road, Arasur

COIMBATORE- 641 407

JUN-2025

KPR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonomous, NAAC 'A')
Avinashi Road, Arasur

COIMBATORE- 641 407

BONAFIDE CERTIFICATE

This is to certify that the **Internship** report submitted by **RAGUL A (Reg No.: 711324321033)** is work done by her and submitted during the academic year 2025 – 2025, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Artificial Intelligence and Data Science**, at **Pinesphere Solutions., Coimbatore.**

Department IIPC Coordinator

Mr Selvakumar G.

Assistant Professor (Sr .G.)

Department of Artificial Intelligence and Data Science

KPR Institute of Engineering and Technology
Arasur, Coimbatore - 641407

Head of the Department

Dr Saranaya N.

Professor and Head

Department of Artificial Intelligence and Data Science

KPR Institute of Engineering and Technology
Arasur, Coimbatore - 641407

Place : Coimbatore
Date :

INTERN COMPLETION CERTIFICATE

PINESPHERE

Pinesphere Solutions
211 - 1st floor, C - block, Dr. N.G.P Institute of Technology
Dr. N.G.P. Nagar, Kalapatti Main Rd, Coimbatore
Tamil Nadu 6410488

TO WHOMESOEVER IT MAY CONCERN

This is to certify that **Mr.RAGUL A** (Reg no: 23AD050), student of **B.Tech Artificial Intelligence and Data Science** from KPR Institution of Engineering and Technology has successfully completed his internship on "**Full Stack Development (Python Django)**" in our concern from **23 June 2025 to 07 July 2025.**

During this period, we found him to be sincere in his work and 100% regular in attendance. We certify that his conduct and character have been good.

Yours Sincerely,

Pinesphere Solutions Pvt. Ltd.

For Pinesphere Solutions

Surendiran S

Director

Date of issue: 11-July-25

ACKNOWLEDGEMENT

First, I would like to thank **Mr. Athithya,** Pinesphere Solutions**.,** for giving me the opportunity to undergo industry training within the organization.

I would also like to thank all the people who worked along with me in **Pinesphere Solutions.,** for their patience and openness in sharing knowledge they created for an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to our Chairman **Dr. Ramasamy K. P.**, our Chief Executive Officer **Dr. Natarajan A. M.,** and our beloved Principal **Dr. Saravanan D.,** for the facilities provided to accomplish this internship.

I would like to thank my Head of the Department **Dr. Saranaya N.,** for her/his supportive actions by permitting me for my internship.

I would like to thank **Dr. Kiruba Shankar R.**, Professor & Head IIPC, KPRIET for his support and advice to get an internship and complete the same in the above said organization.

I am extremely grateful to my department IIPC coordinator **Mr. Selvakumar G.**, faculty members and friends who helped me in successful completion of this internship

S

ABSTRACT

This internship report details my immersive experience at **Pinesphere Solutions**, where I explored the field of **Full Stack Web Development using Python Django** during the period (23/06/2025 – 07/07/2025). Throughout this internship, I actively engaged in learning and implementing both **frontend** and **backend** development technologies to understand the end-to-end workflow of dynamic web applications.

Through hands-on development with **HTML, CSS, and JavaScript** on the frontend and **Python Django** on the backend, I gained valuable skills in UI design, routing, model-view-controller architecture, and database integration. I also worked with **PostgreSQL**, enabling robust data storage and retrieval through Django's ORM system.

The initial phase involved mastering core frontend technologies and setting up the Django environment. As the internship progressed, I built reusable templates, connected pages using Django views, and created models and forms to collect and store user input securely in the PostgreSQL database. I also learned to manage migrations, validate input, and connect the database to the web interface.

The culmination of this learning was a **full-stack portfolio project**, where I implemented features such as:

- Designing responsive web pages using HTML, CSS, and JavaScript
- Creating Django apps and views to handle dynamic content
- Building models and forms to store user input in a PostgreSQL database
- Integrating frontend pages with backend logic and rendering data dynamically

In conclusion, this internship laid a strong foundation for my journey in full stack development. It not only enhanced my technical skills but also sparked a deep interest in building scalable web applications. I look forward to expanding my knowledge in advanced Django development, REST APIs, deployment, and modern JavaScript frameworks.

TABLE OF CONTENTS

CHAPTER 1

About the Company

Pinesphere Solutions is a technology-driven software development company based in Coimbatore, Tamil Nadu, dedicated to nurturing future tech professionals through structured, project-oriented training and mentorship programs. The company specializes in cutting-edge web development technologies, offering immersive learning experiences in areas such as frontend development, backend development using Python Django, database integration, and full stack web application design.

Pinesphere Solutions adopts a hands-on, mentor-guided approach to upskill students, equipping them with both theoretical knowledge and real-world problem-solving abilities. Their internship programs are carefully designed to simulate actual industry workflows, preparing students to meet the demands of modern tech roles.

In addition to technical training, Pinesphere encourages self-paced learning and innovation-driven development, fostering a growth mindset among participants. Through collaborative learning environments, the company builds a strong community of learners and professionals who share knowledge, tools, and experiences to grow together.

Pinesphere Solutions is committed to providing equal opportunities and quality mentorship to students from diverse academic and socio-economic backgrounds. Its mission extends beyond technical training — it aims to empower aspiring developers with the confidence, skills, and mindset required to thrive in a competitive tech ecosystem.

CHAPTER 2

Plan of Your Internship Program

Introduction to Full Stack Web Development:

A comprehensive overview of Full Stack Web Development, focusing on its importance in today's software industry and the demand for developers proficient in both frontend and backend technologies. This internship introduced the fundamental building blocks required to develop dynamic and responsive web applications from scratch using industry-standard tools and frameworks.

The training emphasized the integration of **HTML, CSS, and JavaScript** for frontend development with **Python Django** for backend development. It also included hands-on practice with **PostgreSQL**, enabling efficient data storage and management. The end goal was to equip interns with the skills needed to design, develop, and deploy a fully functional full-stack web application.

Tools and Platforms Used:

The internship program utilized a combination of development tools and platforms to ensure a seamless learning experience:

- **Visual Studio Code (VS Code):** Code editor for writing and organizing frontend and backend code efficiently.
- **Python & Django:** Core backend framework used for creating web applications, handling logic, models, and server-side processing.
- **PostgreSQL:** Relational database system integrated with Django ORM for persistent data storage.
- **HTML, CSS, JavaScript:** Technologies used to design and implement the user interface and ensure responsiveness.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1st WEEK

DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
23/06/2025	Monday	Introduction to HTML5 & CSS3 – Tags, Layout, Styling
24/06/2025	Tuesday	JavaScript Basics – Events, Functions
25/06/2025	Wednesday	Installed Python, Django Framework Setup
26/06/2025	Thursday	Created a New Django Project and Configured Template
27/06/2025	Friday	Created Basic Pages: Home, About, Projects, Contact
28/06/2025	Saturday	Connected Frontend with Django Views and URLs
30/06/2025	Monday	Introduction to Django Models and Forms

2nd WEEK

DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
01/07/2025	Tuesday	Created Model for Contact Form Data
02/07/2025	Wednesday	Built Django Form and Captured Data from Users
03/07/2025	Thursday	Integrated PostgreSQL with Django
04/07/2025	Friday	Migrated Models to PostgreSQL and Stored Form Data
05/07/2025	Saturday	Validated and Cleaned Form Data
06/07/2025	Sunday	Completed Full-Stack Portfolio Website
07/07/2025	Monday	Hosted Project Locally, Tested Routes and Data Storage

CHAPTER 3

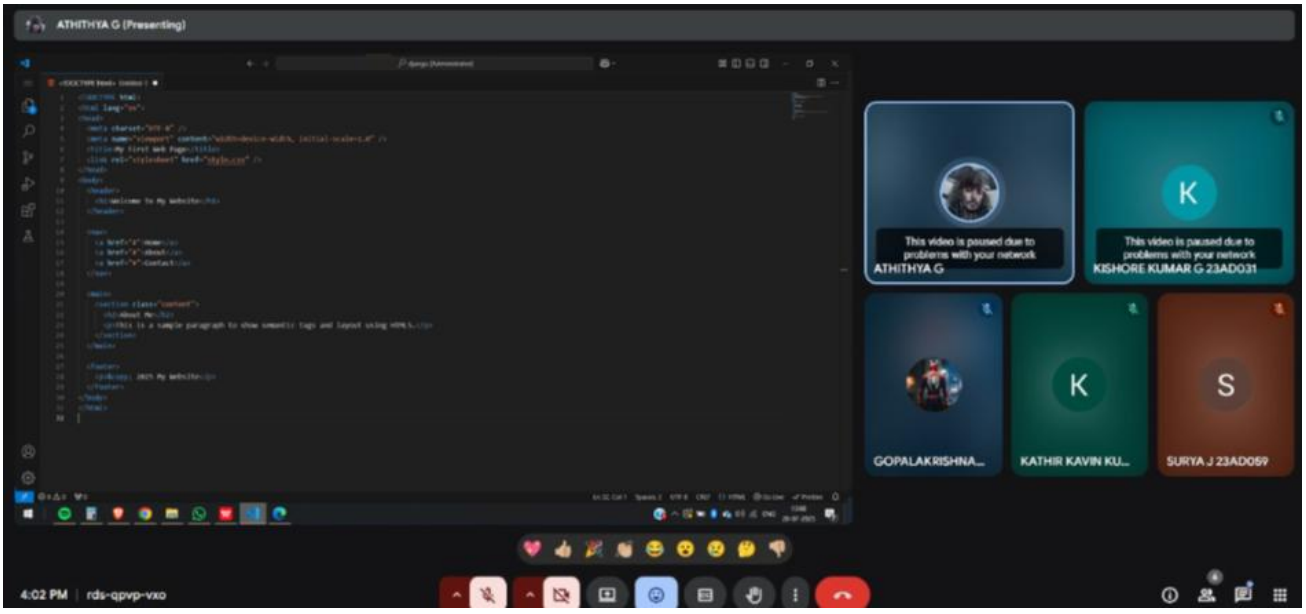
Training Task

Day 1:
Introduction to HTML5 & CSS3 – Tags, Layout, Styling

On the first day, we were introduced to the structure and design of web pages using HTML and CSS.Basics of cloud computing and its impact on businesses.

- Learned about HTML5 elements and semantic tags.
- Created webpage layouts using ** , <header>, <footer>, <nav>**, etc.
- Applied styling using internal and external CSS.
- Explored CSS box model, positioning, and responsive units.
- Hands-on: Designed a basic web page with headers, paragraphs sections.

I practiced structuring a portfolio layout using nested elements and enhanced it with stylings like background colors, padding, and borders. We used internal and external stylesheets, learned best practices like using classes over inline styles, and designed a homepage layout prototype. This foundation was essential for the frontend of the final project.



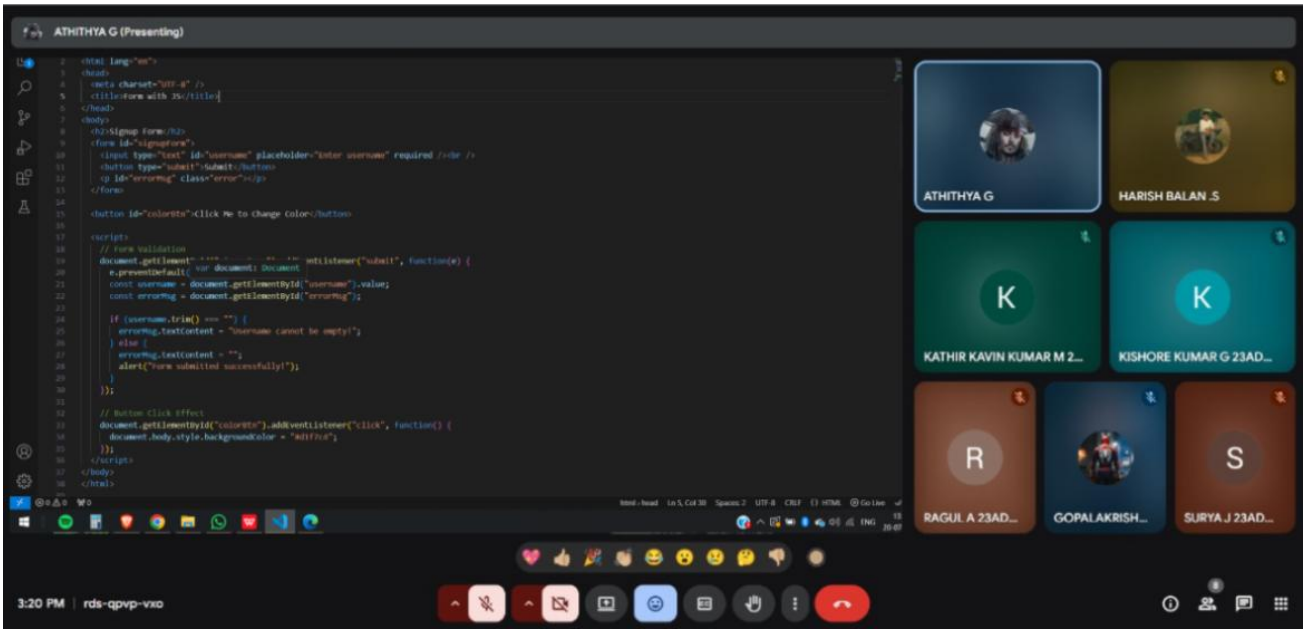
Day 2:

JavaScript Basics – Events, Functions

This session introduced JavaScript for adding interactivity to web pages.

- Covered JavaScript syntax, variables, and data types.
- Learned to use functions and event listeners.
- Manipulated the DOM using getElementById, addEventListener, etc.
- Hands-on: Created a form validation script and interactive button click effects.

I implemented form validation by accessing input values through the **DOM** (getElementById) and displayed messages using innerText. We also explored **browser-based debugging** using console tools. By the end of the session, I could create dynamic web behaviors like hiding/showing elements, validating form fields, and handling events efficiently — all of which helped enhance the UI in the final project.



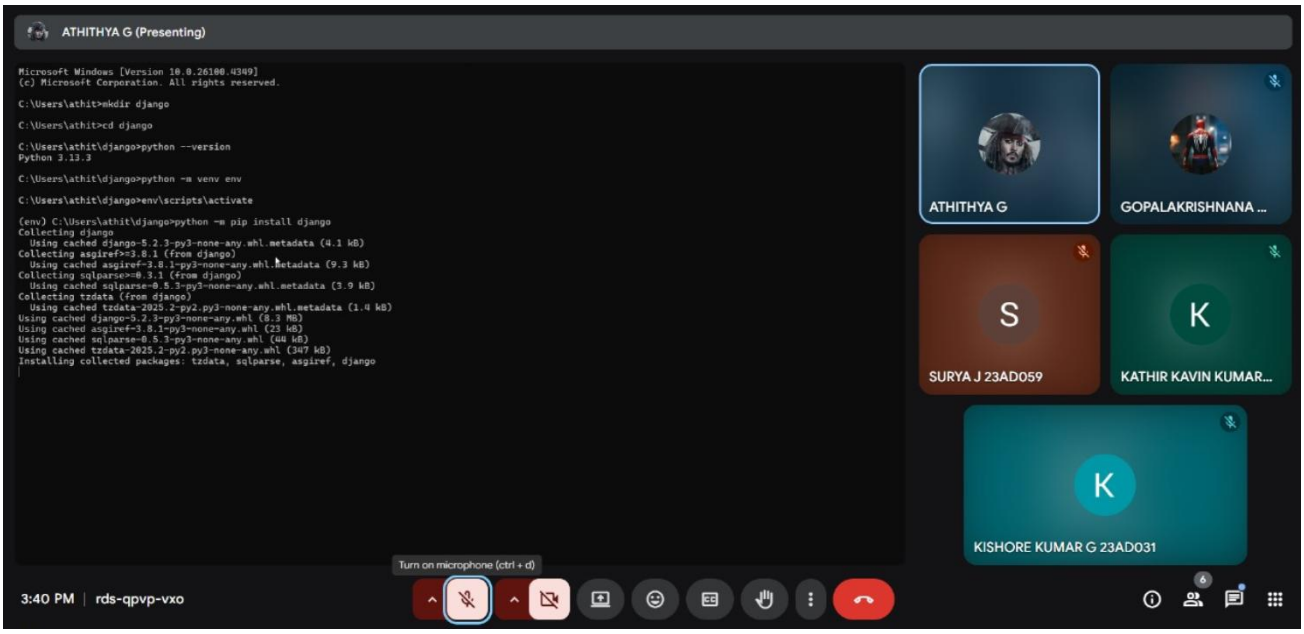
DAY 3:

Installed Python, Django Framework Setup

Focused on setting up the development environment for backend development.

- Installed Python, pip, and Django framework.
- Set up virtual environments for Django projects.
- Hands-on: Created a new Django project using **django-admin startproject.**
- Explored Django project structure and purpose of **manage.py.**

We explored how Django uses the **MTV architecture (Model-Template-View)** and compared it with the traditional MVC pattern. This session laid the groundwork for full-stack integration by enabling a Python-powered backend that interacts with HTML/CSS frontends and databases.



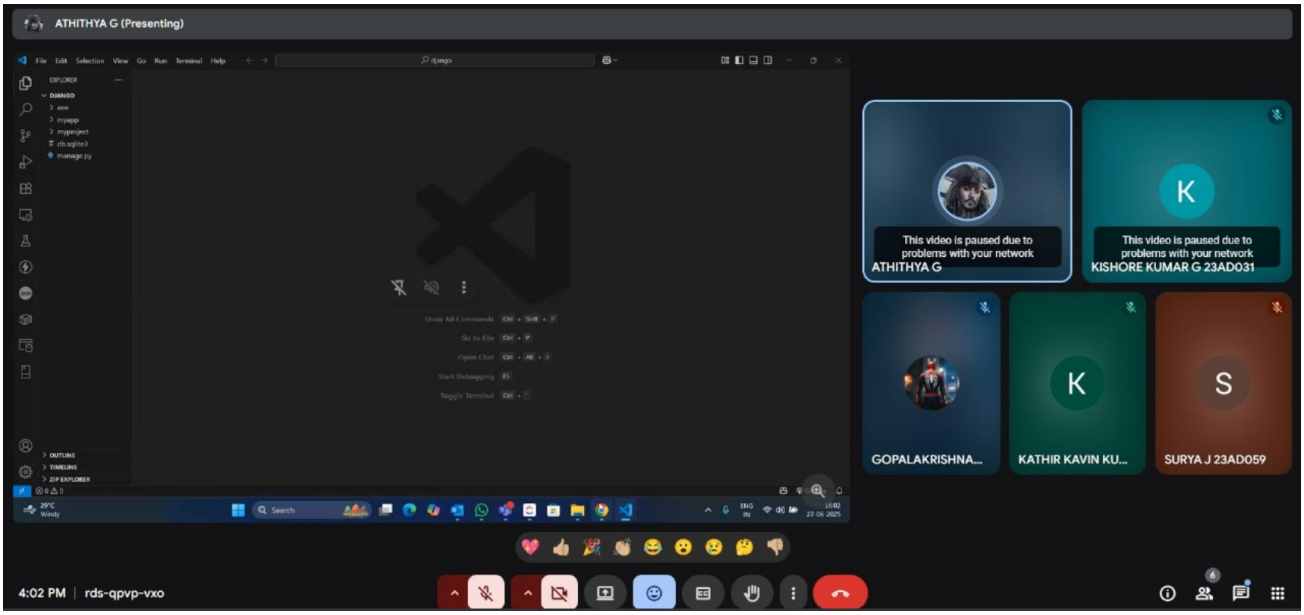
DAY 4:

Created a New Django Project and Configured Template

We built our first Django app using startapp, added it to INSTALLED_APPS, and registered URL paths using Django's routing system. I learned how to organize frontend files (HTML, CSS, JS) using Django's STATICFILES_DIRS and TEMPLATES settings

- Created a new Django app and registered it in **settings.py.**
- Learned about views, URLs, and templates.
- Configured static files and template directories.
- Hands-on: Rendered HTML templates using Django's templating engine.

Using **Django's template engine**, I rendered HTML files and introduced **template tags** like {% block %}, {% extends %}, and {% static %}. This helped in setting up reusable templates — key to reducing code duplication in the final project.



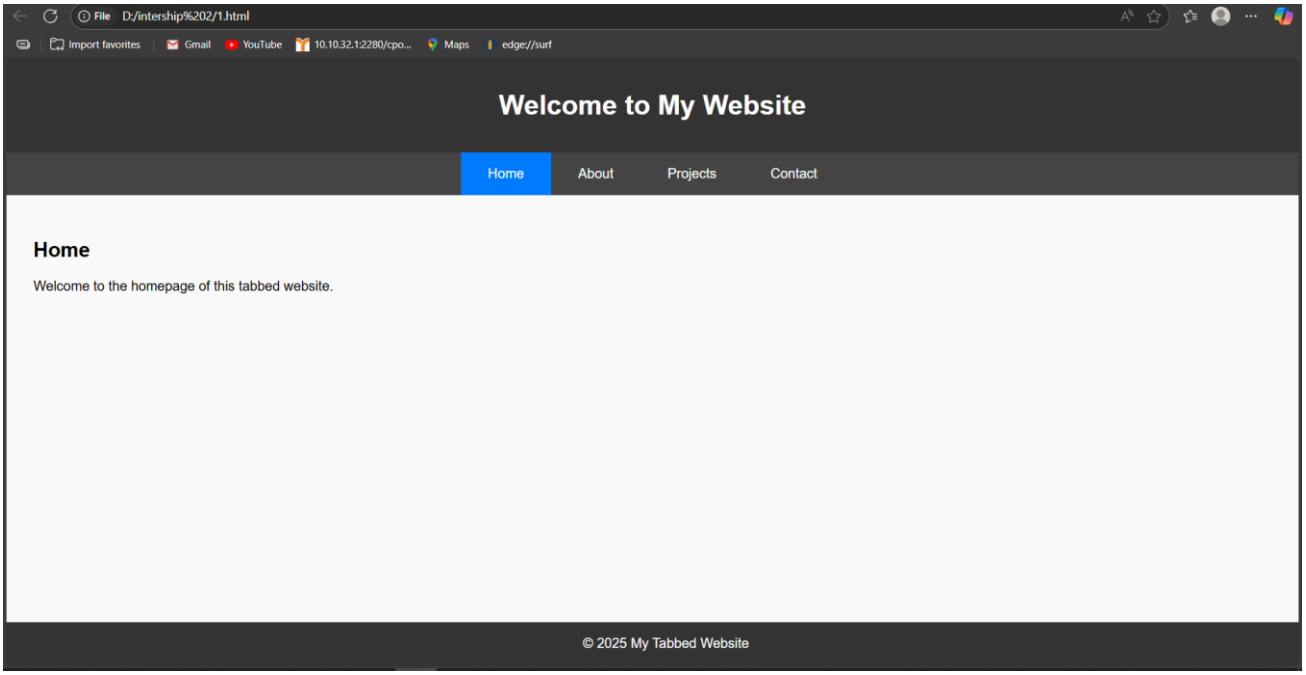
DAY 5:

Created Basic Pages: Home, About, Projects, Contact

On Day 5, I created multiple views using Django functions and mapped them with their respective URLs. Each view was linked to a corresponding HTML file — styled and structured using previously learned CSS techniques. Pages were designed using **semantic HTML**, emphasizing accessibility and readability

- Created multiple views and mapped them to URLs.
- Designed page layouts for different sections of a portfolio website.
- Added navigation links between pages.
- Hands-on: Built a responsive navigation bar and footer.

I implemented a responsive **navigation bar** and **footer** that appeared consistently across pages using template inheritance. These sections would later be expanded to support dynamic content and forms.



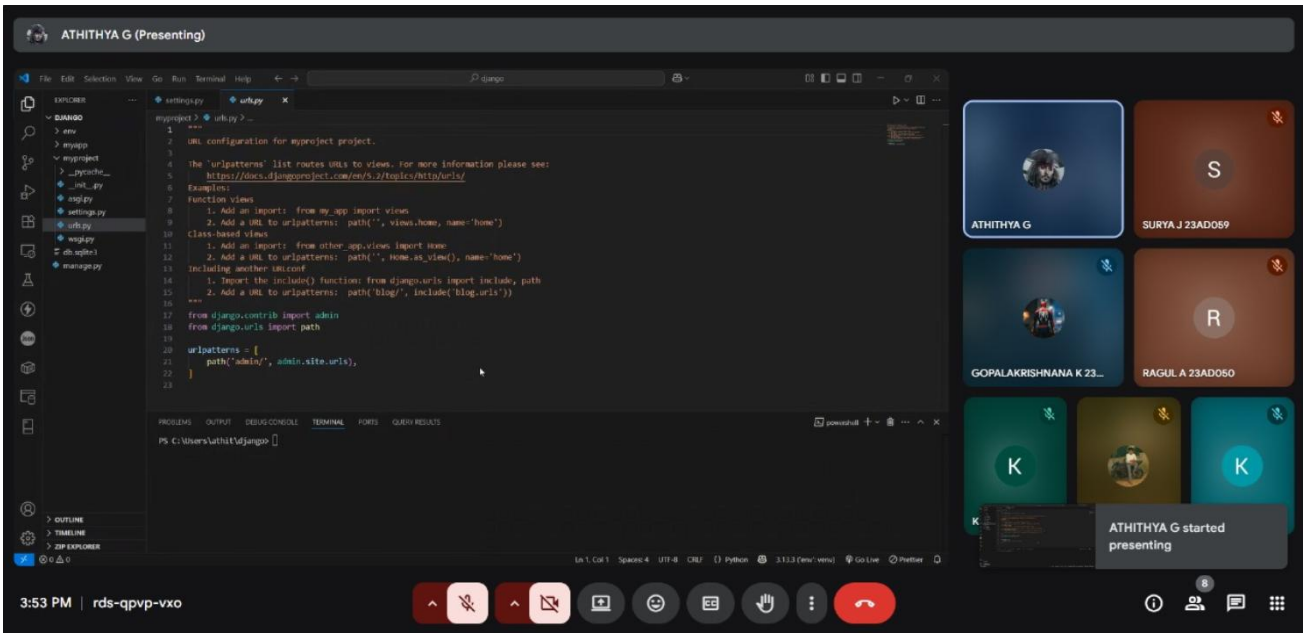
DAY 6:

Connected Frontend with Django Views and URLs

I established links between the user interface and backend logic using **Django views and context data**. The focus was on creating dynamic HTML responses using render() and passing data (dictionaries) to templates.

- Connected HTML pages dynamically using Django views.
- Used Django template inheritance (index.html).
- Rendered content conditionally using template tags.
- Hands-on: Displayed dynamic data using Django context variables.

I used **template inheritance** (index.html) to modularize page layouts and rendered dynamic messages like welcome text and live data. Conditional statements ({% if %}, {% for %}) were used to loop through data, an essential step in listing projects or form entries dynamically.



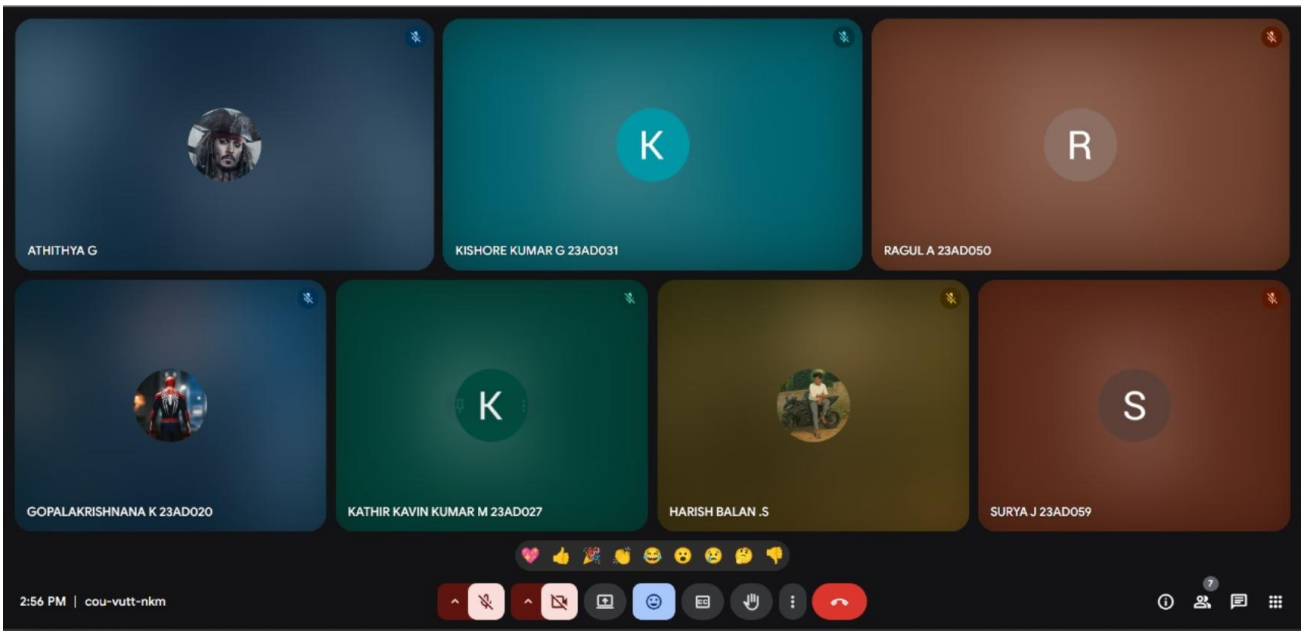
DAY 7:

Introduction to Django Models and Forms

This session introduced the **Object-Relational Mapping (ORM)** system. I created models that represented tables in the database using Django's models.Model class. Each field (like CharField, EmailField, TextField) mapped to a column in the database.

- Learned about Django ORM and how models represent database tables.
- Created model classes with fields and validations.
- Introduced to Django Forms for handling user input.
- Hands-on: Designed a basic contact form with name, email, and message fields.

I then explored **Django Forms**, which allow server-side validation and form rendering. We discussed CSRF protection, form submission, and error messaging. This gave us the ability to take user input securely — a central part of the portfolio's contact page.

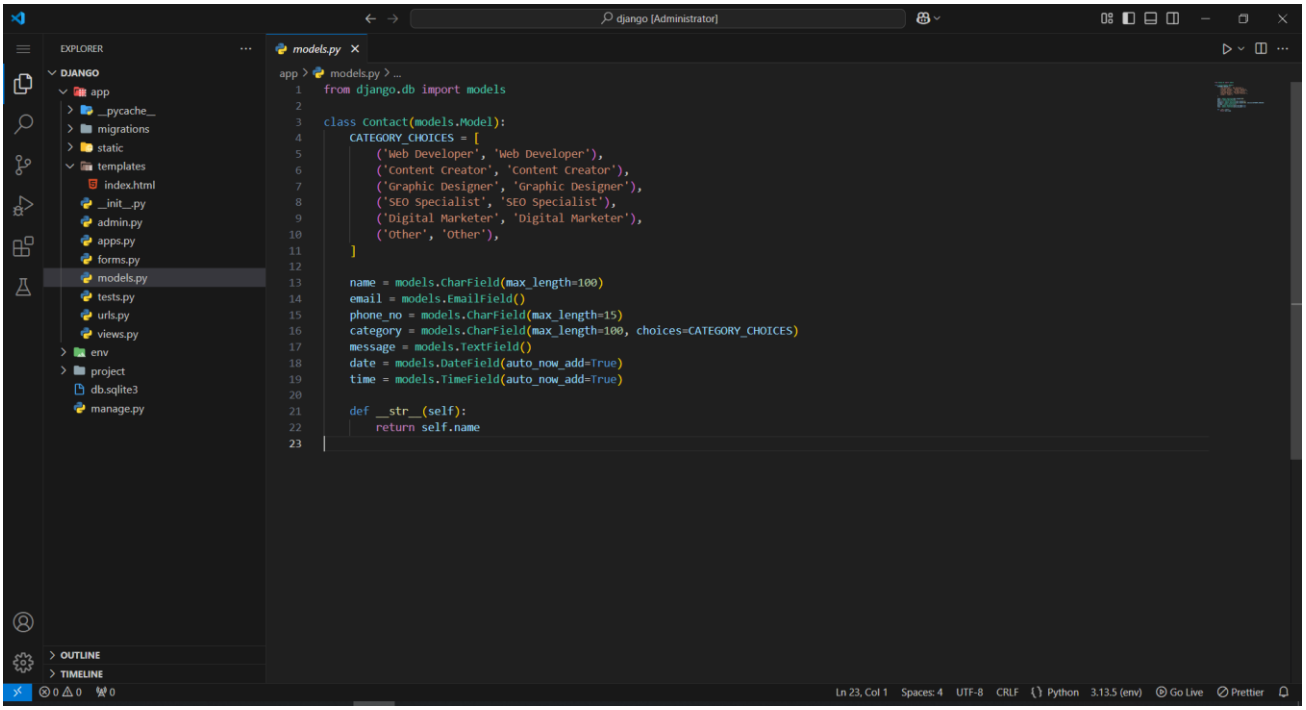


DAY 8:
Created Model for Contact Form Data

On this day, I designed a model named Contact, consisting of fields like name, email, and message. I added metadata like verbose_name and defined __str__() methods for human-readable output in Django Admin.

- Created a model to store user inputs from the contact form.
- Added metadata, verbose names, and string representation methods.
- Hands-on: Generated and applied migrations using **makemigrations** and **migrate.**

I used Django's migration system to create database tables (makemigrations, migrate). These migrations ensured consistent schema updates without manual SQL queries. This model was the base for the form used on the Contact page.



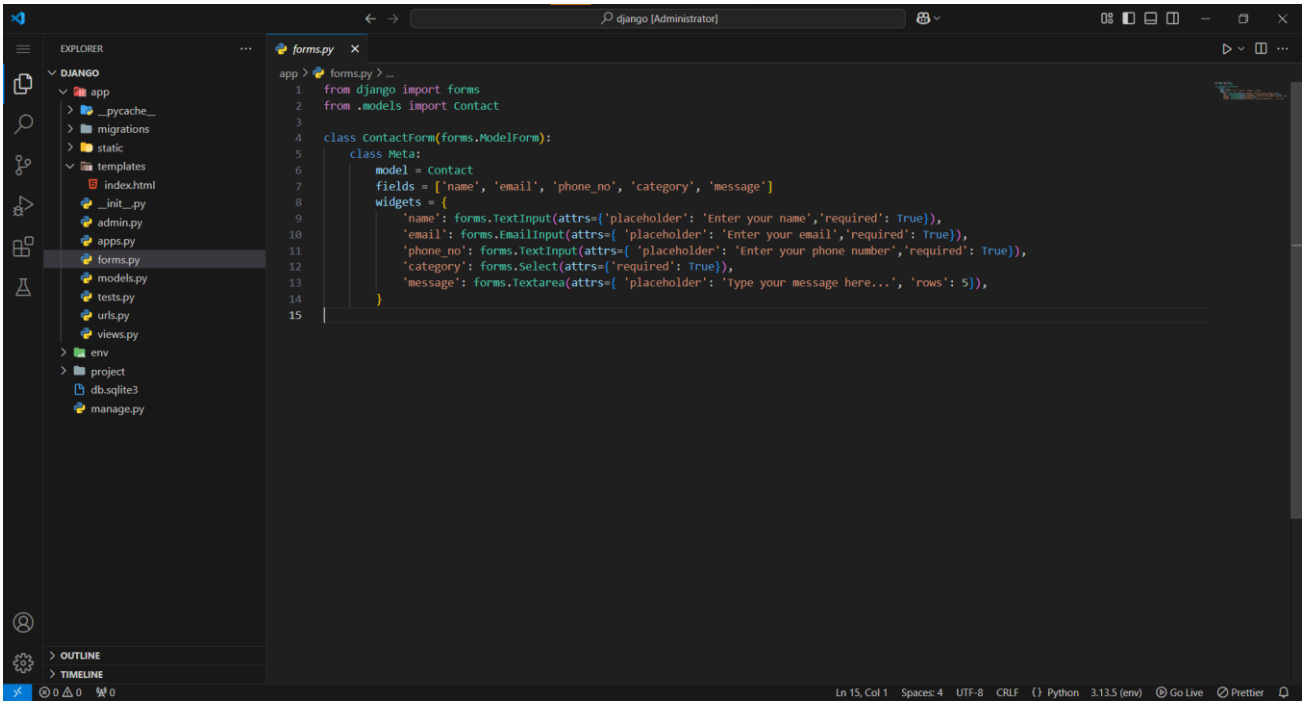
DAY 9:

Built Django Form and Captured Data from Users

I used Django's ModelForm class to create a reusable form linked to the Contact model. The form was rendered in the frontend using {% csrf_token %} for security and auto-populated with Django-generated fields.

- Built a Django ModelForm to simplify form creation.
- Rendered form fields using Django templating.
- Captured and processed user-submitted form data.
- Hands-on: Stored data to the database and displayed a confirmation message.

After form submission, I wrote logic to validate input (form.is_valid()) and save data to the database. On success, a “Thank you” message appeared using redirect or render logic with context variables. This process mimicked real-world form handling and backend data storage in production systems



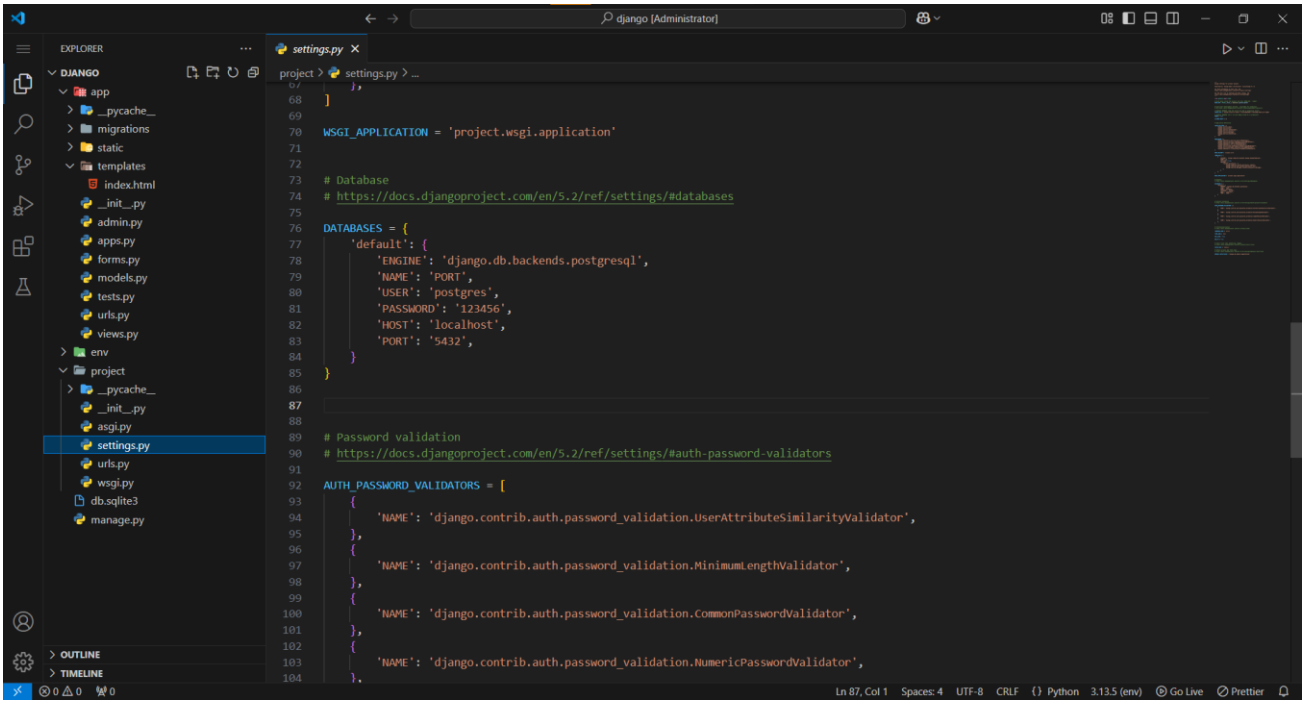
DAY 10:

Integrated PostgreSQL with Django

We replaced Django's default SQLite database with **PostgreSQL**, a more scalable and production-ready RDBMS. After installing psycopg2, I updated the DATABASES setting in settings.py with user credentials.

- Installed **PostgreSQL** and **psycopg2** driver.
- Configured database settings in Django **settings.py.**
- Created a new PostgreSQL database for the project.
- Hands-on: Tested connection between Django and PostgreSQL

A new database was created in PostgreSQL using pgAdmin, and Django was connected to it seamlessly. I ran the necessary migrations to create tables in PostgreSQL and tested connectivity using Django's ORM methods.



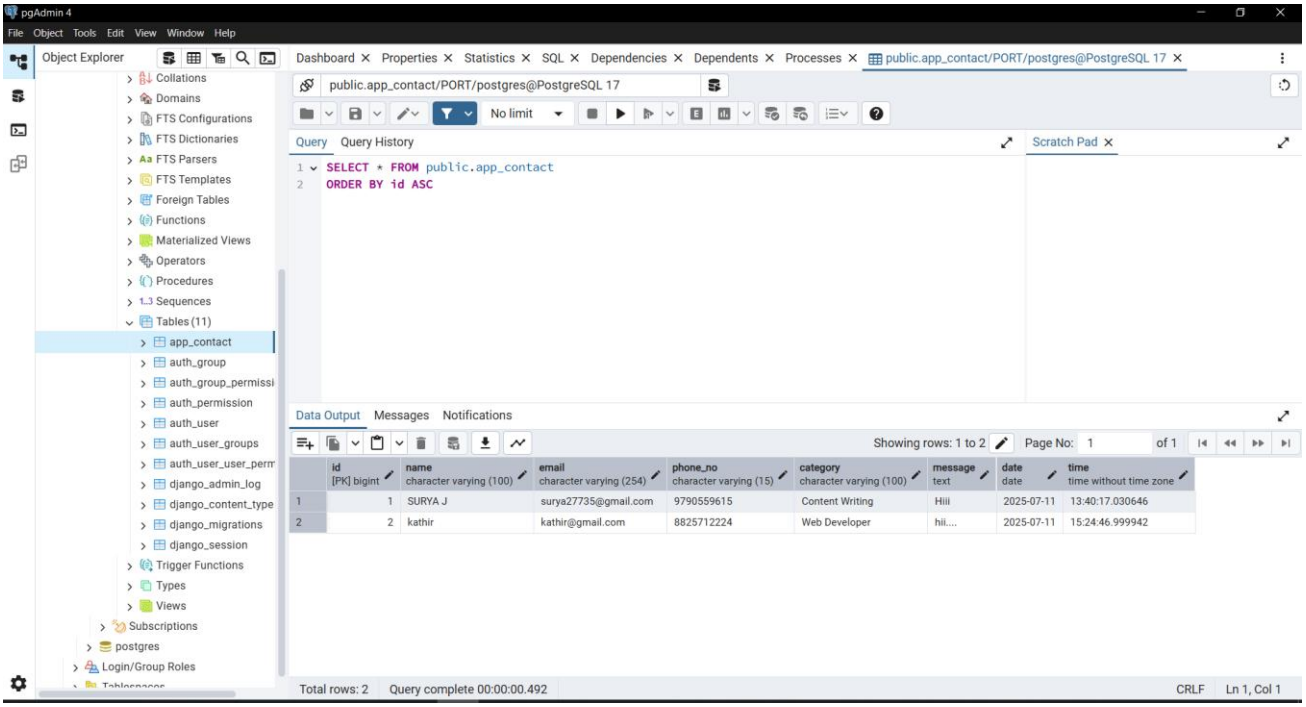
DAY 11:

Migrated Models to PostgreSQL and Stored Form Data

Using PostgreSQL as the backend, I ran all previously created migrations to build the model schema inside the new database. I explored how Django creates and manages tables in PostgreSQL and confirmed data storage through SQL queries and admin panels.

- Applied existing migrations to the PostgreSQL database.
- Verified table structure using pgAdmin or SQL queries.
- Tested form submissions and confirmed data was being stored successfully.
- Hands-on: Retrieved and displayed data using Django querysets.

Form submissions were tested end-to-end, verifying that new entries were successfully written and retrieved from the database. This exercise taught me how Django abstracts SQL queries using Python objects



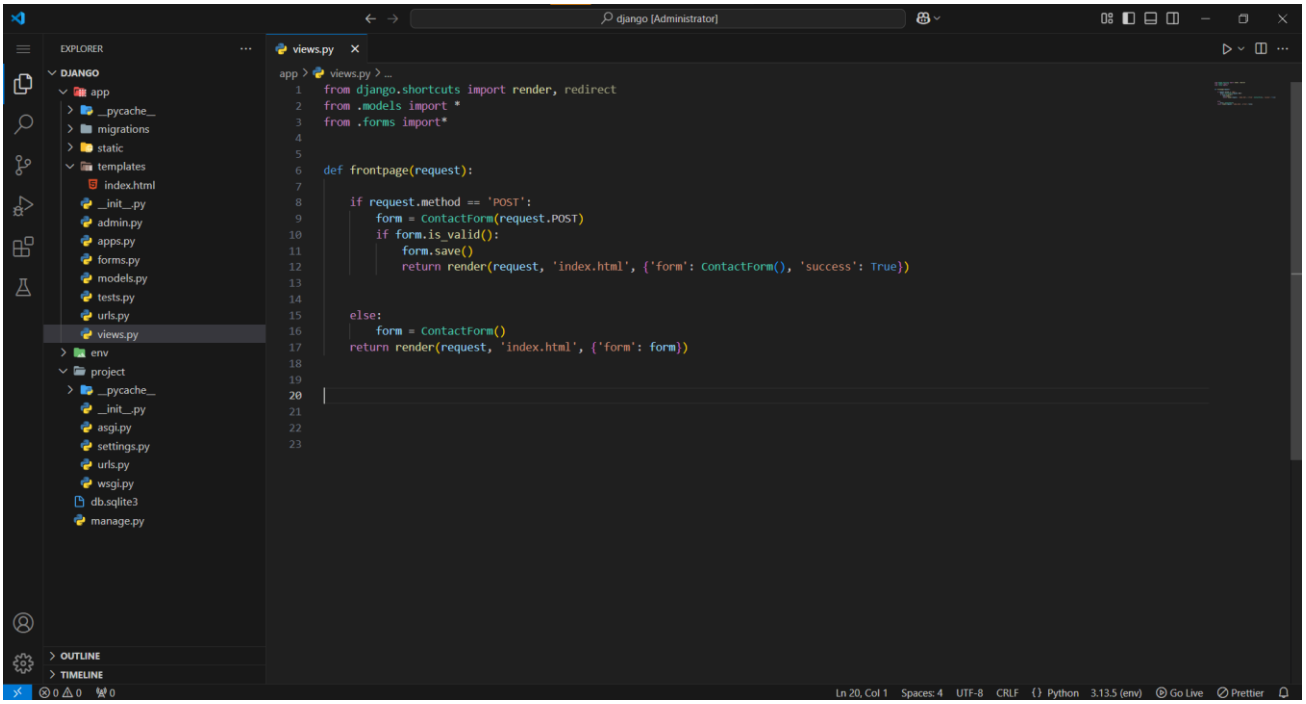
DAY 12:

Validated and Cleaned Form Data

I implemented Django's form validation system to clean data before saving it. Using the clean() method and custom validators, I ensured inputs followed required rules (e.g., non-empty name, valid email format, length restrictions).

- Implemented Django form validations and custom error messages.
- Learned about the **clean()** and **is_valid()** methods.
- Prevented duplicate or invalid submissions.
- Hands-on: Displayed user-friendly validation feedback on the frontend.

User feedback was improved with conditional rendering of error messages. This introduced me to the concept of **server-side validation**, an important layer of security in real-world applications.



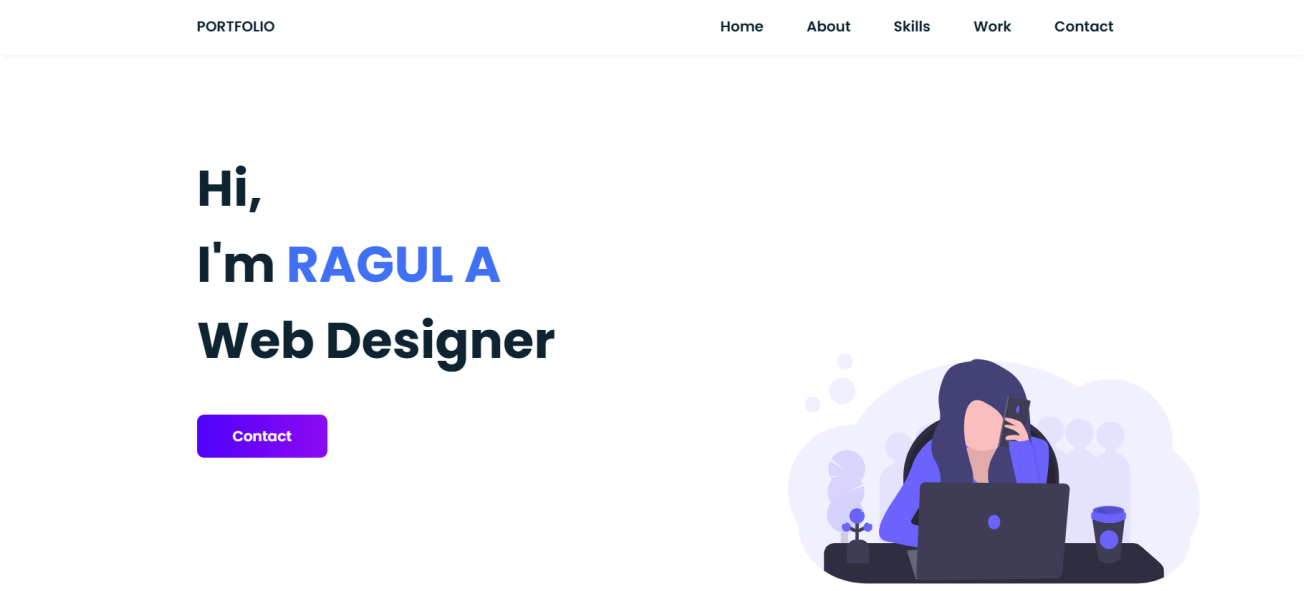
Day 13:

Completed Full-Stack Portfolio Website

This day was focused on final integration and polishing. I finalized layouts, styled all frontend sections using responsive CSS, and checked for mobile compatibility using dev tools.

- Finalized styling and layout for all pages.
- Linked contact form to backend with successful submission messages.
- Organized static files (CSS, JS, images) and ensured all paths worked.
- Hands-on: Polished and reviewed the portfolio site as a complete full-stack project.

Navigation between views was tested, and success messages were enhanced with alert boxes. All static assets (CSS, JS, images) were properly configured and loaded. The result was a complete, visually appealing, and fully functional **full-stack portfolio**.



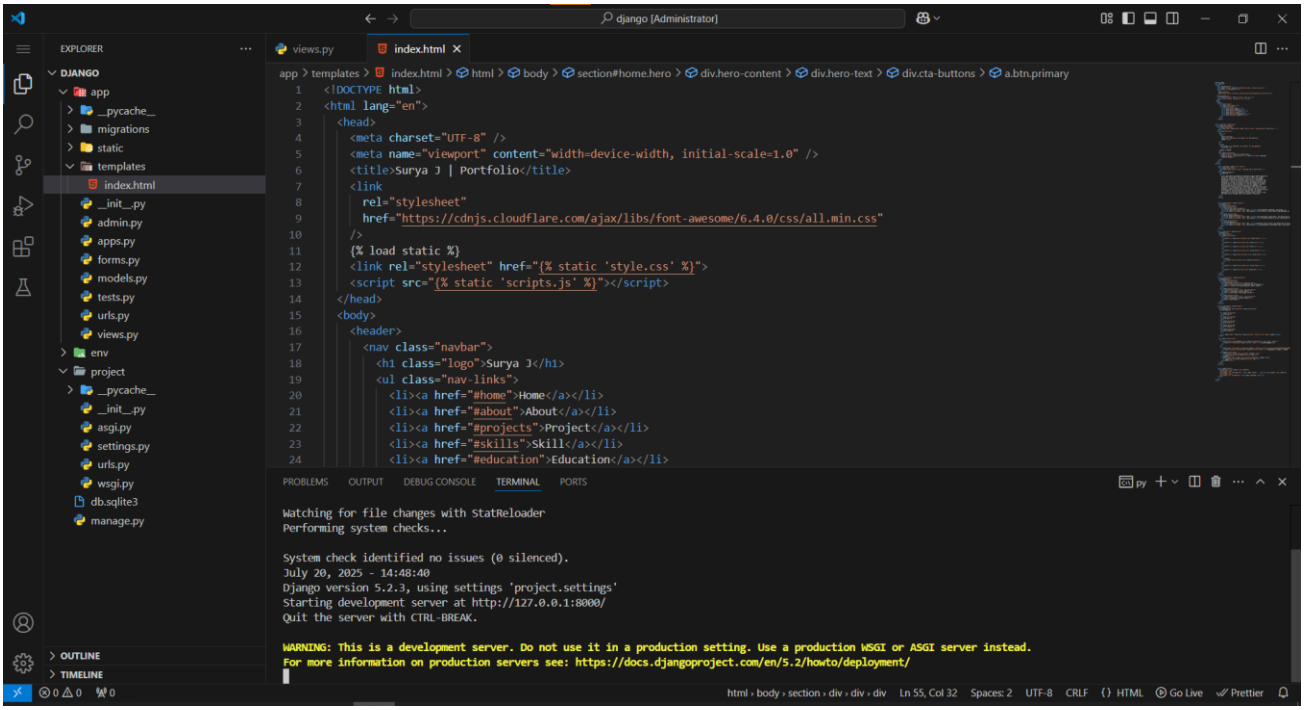
Day 14:

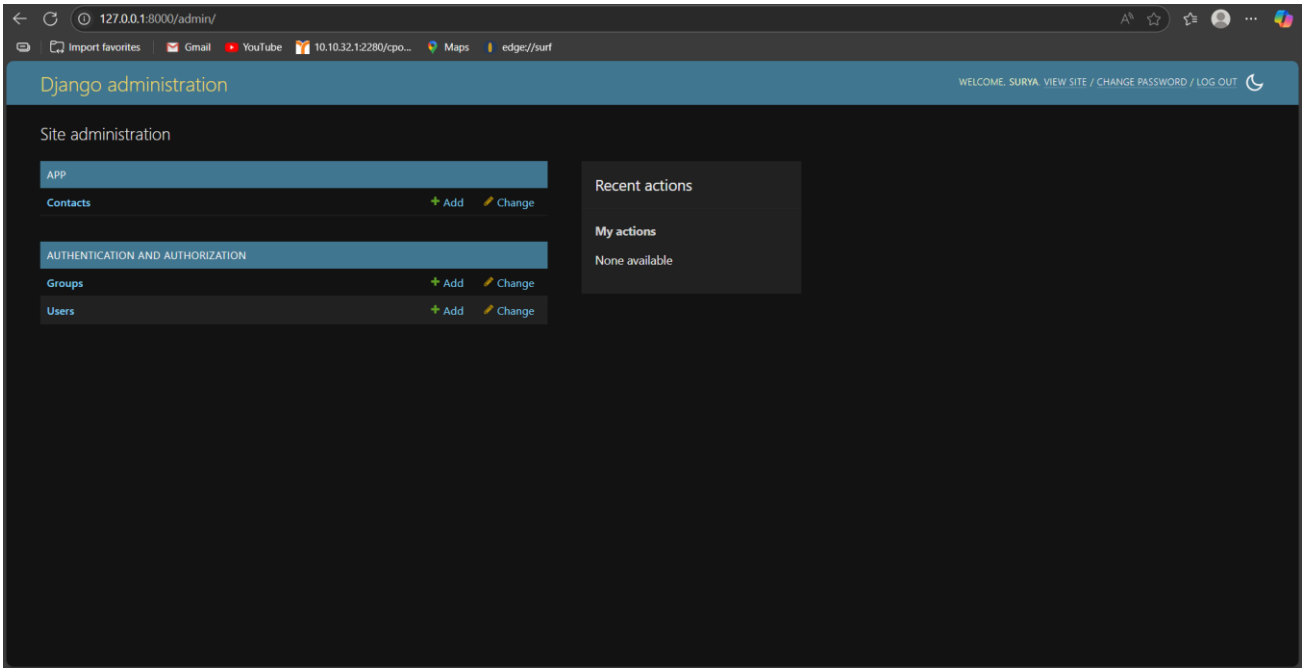
Hosted Project Locally, Tested Routes and Data Storage

I ran the Django development server and tested all endpoints (http://127.0.0.1:8000/). Functionality of pages like Home, Projects, and Contact was verified, including backend database operations.

- Ran the Django server and tested all routes for correct responses.
- Checked database operations (Create, Read) from frontend to backend.
- Debugged minor issues in URL routing, form handling, or template paths.
- Successfully hosted the project on localhost and demonstrated full functionality.

Access to Django Admin (http://127.0.0.1:8000/admin/) allowed data management and model entry confirmation. Debugging was done using logs and browser tools. This final step helped simulate deployment preparation and introduced best practices for local testing before going live.




127.0.0.1:8000/admin/
Import favorites
Gmail
YouTube
10.10.32.1:2280/cpo...
Maps
edge://surf
Django administration
WELCOME, SURYA. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
APP
Contacts
Add
Change
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
Recent actions
My actions
None available

ATTENDANCE SHEET

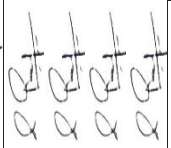
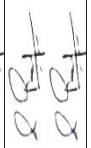
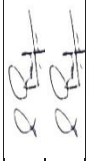
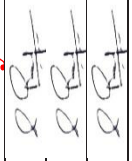
PINESPHERE SOLUTIONS

211-1st floor, c-lock, Dr.NGP.Institute of Technology, Dr.N.G.P.Nagar, Kalapatti Rd,

Coimbatore, Tamil Nadu 6410488

Name of Student	RAGUL A
Roll. No	23AD050
Title of Intern	Full Stack Development (Python Django)
Date of Commencement of Training.	23-06-2025
Date of Completion of Training:	07-07-2025

Initials of the student

Month & Year	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
June 2025																												Holiday		
July 2025				Holiday																											

Note:

1. Attendance Sheet should remain affixed in Daily Training Diary. **Do not remove or tear it off.**
2. Students should sign/initial in the attendance column. Do not mark 'P'
3. Holidays should be marked in **Red Ink** in the attendance column. Absent should be marked as **'A' in Red Ink.**

Signature of Industry Supervisor with

company stamp/seal

(Name_____) Contact No.

INDUSTRY SUPERVISOR EVALUATION OF IN-PLANT TRAINING / INTERNSHIP / INDUSTRY PROJECT WORK

Name of the Student: RAGUL A Date: 07.07.2025

Name of Industry Supervisor with Designation: Mr. Athithya

Company/Organization: Pinesphere Solutions

Address: 211-1st floor, c-lock, Dr.NGP.Institute of Technology, Dr.N.G.P.Nagar, Kalapatti Rd,Coimbatore, Tamil Nadu 6410488

Dates of Internship: From 23-06-2025 To 07-07-2025

Please evaluate your intern by indicating the frequency with which observed behaviors:

Parameters	Needs improvement	Satisfactory	Good	Excellent
Behaviors				
Performs in a dependable manner				
Cooperates with co-workers and supervisors				
Shows interest in work				
Learns quickly				
Shows initiative				
Produces high quality work				
Accepts responsibility				
Accepts criticism				
Demonstrates organizational skills				
Uses technical knowledge and expertise				
Shows good judgment				
Demonstrates creativity/originality				
Analyzes problems effectively				
Is self-reliant				
Communicates well				
Writes effectively				
Has a professional attitude				
Gives a professional appearance				
Is punctual				
Uses time effectively				

Overall performance of student intern (circle one):
(Needs improvement/ Satisfactory/Good/Excellent)

Additional comments, if any:

Industry supervisor Signature (with Seal)

CHAPTER 5

Conclusion

This report has explored the essential components of **Full Stack Web Development**, focusing on the practical implementation of frontend technologies like **HTML, CSS, and JavaScript**, combined with powerful backend development using **Python Django** and **PostgreSQL**. Through hands-on experience, I gained valuable insights into building dynamic web pages, integrating user interfaces with backend logic, managing databases, and structuring a complete portfolio website.

Django, as a high-level Python framework, provided a robust and scalable environment to manage data models, handle routing, validate user inputs, and connect seamlessly to a PostgreSQL database. The use of HTML and CSS ensured the development of responsive and user-friendly interfaces, while JavaScript added interactivity to enhance user engagement.

My internship at **Pinesphere Solutions** has been an invaluable experience, offering me a comprehensive understanding of full stack development and the workflow behind real-world web applications. This journey strengthened my technical foundation, improved my problem-solving skills, and deepened my appreciation for clean code and scalable application design.

Overall, this experience has ignited a strong passion for web development and motivated me to further explore advanced frameworks, deployment strategies, and API integrations. I look forward to applying these skills to future projects and continuing my growth as a full stack developer.