VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belgavi, Karnataka State, India

A report on

"Internship"
At
"Apponix Technologies"

Submitted in partial fulfillment of the degree of

Master of Computer Applications

Submitted by

Mr. Chetan Mallappa Melavanki

USN: 2BA23MC008

Under the Guidance of

Prof. M. H. Shirur

Dept. of MCA

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS (MCA)

BASAVESHWAR ENGINEERING COLLEGE, BAGALKOTE

2024-2025

Visvesvaraya Technological University

Belgavi, Karnataka State, India

Basaveshwar Engineering College

Department of Master of Computer Applications (MCA)

Bagalkote – 587 103

CERTIFICATE

This is to certify that **Mr. Chetan Mallappa Melavanki**, bearing **USN:2BA23MC008**, has satisfactorily completed the **'Internship' at Apponix Technologies**, **and** submitted report to the Basaveshwar Engineering College, Bagalkote, in partial fulfillment of the requirements for the award of Master of Computer Applications (MCA), during the academic year 2024-2025.

Prof. M. H. Shirur	**Miss. Shrilakshmi Kulkarni**
Internal Guide	**External Guide**

Prof. S. S. Gujarathi	**Prof. Sudha. K. S**	**Dr. B. R. Hiremath**
Internship Coordinator	**Head of the Department**	**Principal**

Date Of Submission:

Examiners:

1. **2.** **3.**

Approval letter provided by company

Apponix Technologies Pvt. Ltd

Date: 03th December 2024

Name: CHETAN MALLAPPA MELAVANKI

RE: OFFER FOR INTERNSHIP

Following your application and subsequent interview, we are pleased to inform you that you have been considered for internship as a Trainee-Python Full Stack Development in our company, with effect from 04/12/2024 at Hubli Branch.

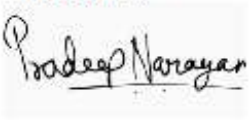
The terms of engagement are as follows:

1. Period of Engagement: 8 weeks
2. Notice period: 1 week from either side.
3. Stipend : Rs 0/- per month

You will be working as Intern Trainee - Python Full Stack Development

It is my hope and prayer that you will work to your level best to improve the efficiency and performance of this company. Congratulations and best wishes.

Yours faithfully



Pradeep Narayan
Managing director
Corporate Office:
306, 10th Main, 46th Cross,
4th Block Rajajinagar
Bangalore – 560010
info@apponix.com
www.apponix.com

Certificate of completion provided from organization

APPONIX academy
SINCE 2013

Date: 30th January 2025

Name: CHETAN MALLAPPA MELAVANKI

TO WHOM IT MAY CONCERN

This is to certify that Mr. **CHETAN MALLAPPA MELAVANKI** has successfully completed his internship in Organization Study at **Apponix Technologies Pvt. Ltd.** from **4th December 2024 to 29th January 2025**.

His **USN Number** is **2BA23MC008**, and he is a student of the **Master of Computer Applications (MCA) Department** at **Basaveshwar Engineering College, Bagalkote**.

As part of the internship, he was included in various Apponix client projects.

During the internship he demonstrated good skills with a self- motivated attitude to learn new things. His performance exceeded expectations and was able to complete the tasks on time.

We wish him all the very best for a successful career ahead. It is my hope and prayer that you will work to your level best to improve the efficiency and performance of this company.

Congratulations and best wishes.

Authorized Signatory

Pradeep Narayan

Pradeep Narayan
Managing director

Corporate Office:
306, 10th Main, 46th Cross, 4th Block Rajajinagar Bangalore – 560010 info@apponix.com
www.apponix.com

DECLARATION

I, **Chetan Mallappa Melavanki** (USN: **2BA23MC008**), a student of **3rd Semester MCA** at **Basaveshwar Engineering College (Autonomous), Bagalkot**, hereby declare that the internship work titled **"Intern Trainee: Python Full Stack Development"** has been carried out **independently by me** at the **Department of Master of Computer Applications**, Basaveshwar Engineering College (Autonomous), Bagalkot.

This internship report is submitted in **partial fulfillment** of the requirements for the award of the degree **Master of Computer Applications (MCA)** by **Basaveshwar Engineering College (Autonomous), Bagalkot**, during the **academic year 2024–2025**.

I have undertaken this internship as an **Intern Trainee – Python Full Stack Development** at **Apponix Technologies, Hubli**, and the work presented in this report is a **genuine record** of the internship carried out by me under the guidance and support of the faculty and industry professionals.

Yours Sincerely,

Chetan Mallappa Melavanki

USN: 2BA23MC008

Acknowledgment

I am profoundly grateful to **Apponix Technologies** for providing me with the opportunity to undertake an 8-week internship on **Python Full Stack Development**, which has been an invaluable learning experience. The knowledge and skills I have gained during this internship will greatly contribute to my professional growth.

I am deeply indebted to **Dr. B. R. Hiremath,** Principal of Basaveshwar Engineering College, Vidyagiri, Bagalkote, for his encouragement and for providing me with the platform to pursue this internship.

I would like to express my sincere gratitude to **Prof. Sudha K. S.**, Head of the Department, for her continuous guidance, support, and motivation throughout my academic journey and during this internship.

I extend my heartfelt thanks to my external guide, **Miss. Shrilakshmi Kulkarni**, for her constant guidance, encouragement, and constructive feedback throughout the internship. Her mentorship was crucial in successfully completing my tasks and projects.

I also wish to thank my internal guide, **Prof. M. H. Shirur**, for his valuable advice, support, and encouragement during this internship. His insights have been instrumental in enhancing my learning experience.

Additionally, I am thankful to our internship coordinator, **Prof. S. S. Gujarathi**, for his efforts in coordinating this internship program and for his continuous assistance throughout the process.

Lastly, I would like to thank my **family**, **friends**, and the **faculty** of my institution for their unwavering support and encouragement, which motivated me throughout this journey.

This internship has been an enriching experience, and I am confident that the knowledge and skills I have acquired will prove invaluable in my future endeavors.

Yours Sincerely,

Chetan Mallappa Melavanki

USN: 2BA23MC008

TABLE OF CONTENTS

1. Introduction to the Organization

Apponix Technologies is a well-established and reputed training institute dedicated to providing high-quality professional skill development programs. Founded in 2013, Apponix has emerged as a trusted name in technical education, offering a wide range of career-oriented courses including IT certifications, digital marketing, data science, cloud computing, and full-stack development. With its headquarters in India and a global presence extending to the USA and the UK, the institute is committed to bridging the gap between academic learning and industry requirements.

The **Hubli branch** of Apponix Technologies, located centrally in the Hubli-Dharwad region, plays a vital role in catering to the skill development needs of students, graduates, and working professionals in North Karnataka. This branch is equipped with modern infrastructure, experienced industry trainers, and a practical, hands-on curriculum aligned with current market trends and employer expectations. The Hubli center aims to empower individuals with the technical skills and confidence required to succeed in today's competitive job market.

2. Organization Profile

Name: Apponix Technologies Pvt. Ltd.

Year Established: 2013

Headquarters: Bangalore, Karnataka, India

Hubli Branch Address: 2nd floor, Virupaksha Krupa building P.B. Road, Vidyanagar, Abover Airtel Store, Opp KMC Main Gate Hubli

Website: https://www.apponix.com

Services Offered

Apponix Technologies is a leading professional training institute offering career-oriented programs in:

- Full Stack Web Development
- Digital Marketing
- Data Science
- Cloud Computing
- Networking & Cybersecurity
- DevOps
- Corporate Training & Career Counseling
- Placement Support Services

Special Features

- **Flexible Learning Options:** Both classroom and online training formats available.
- **Experienced Faculty:** Trainers with over 6+ years of industry and teaching experience.
- **Placement Assistance:** Includes resume writing, mock interviews, and job referrals.
- **Practical Curriculum:** Focus on hands-on training and real-time projects.
- **Affordable Fees:** Competitive pricing for high-quality training.
- **Updated Course Content:** Designed as per the current industry needs.

Mission Statement

"To empower students and professionals with industry-relevant skills that enable them to achieve their career goals."

Vision

"To be a global leader in skill development and technical education by delivering innovative, job-oriented training programs and personalized support."

Hubli Branch Overview

The **Hubli branch** of Apponix Technologies caters specifically to the talent needs of the **Hubli-Dharwad region**, offering world-class training in a locally accessible environment. It is equipped with modern labs, fast internet, and industry-grade tools to deliver a smooth learning experience. The trainers here offer personalized mentoring and help students transition from learning to employment with confidence.

Internship Role: Intern Trainee – Python Full Stack Development

During the internship, we worked as **Intern Trainees** specializing in **Python Full Stack Development**. The training focused on the following technologies:

- **Frontend:** HTML, CSS, JavaScript, Bootstrap
- **Backend:** Python with Django framework
- **Database:** MySQL
- **Version Control:** Git

We gained practical experience by working on dynamic web projects, including CRUD operations, user interfaces, form handling, and basic authentication systems. The internship emphasized real-world problem-solving, best coding practices, and teamwork. This experience laid a solid foundation for building full-stack web applications.

3. Discussion on Departments Studied

1. Training & Development Department

This is the primary department responsible for delivering technical training programs. It manages:

- Course content design and updates
- Trainer allocation and scheduling
- Hands-on training and assessments
- Mentoring and project guidance

Intern Role: We worked under this department as Python Full Stack Development interns, participating in hands-on sessions and real-time project work.

2. Student Support & Counseling Department

This team handles:

- Student registration and onboarding
- Batch coordination and scheduling
- Counseling students regarding learning paths
- Handling grievances and feedback

Intern Interaction: They guided us with schedule updates, attendance, and general inquiries during our internship.

3. Placement & Career Services Department

This department supports:

- Resume building, mock interviews, and soft skills training
- Sharing job openings and interview opportunities
- Coordinating with partner companies for placements

Intern Interaction: Although not directly involved in placements, we were briefed on how this department prepares students for job readiness.

4. Marketing & Admissions Department

This department manages:

- Promotions through online and offline campaigns
- Responding to inquiries via calls, emails, and website
- Enrolling students in appropriate courses
- Collaborating with colleges or industries for training programs

Intern Observation: We observed how this team plays a crucial role in reaching students and maintaining the institute's reputation.

5. Administration & Operations Department

Responsible for:

- Daily operations, classroom setup, and lab maintenance
- Coordinating with trainers and departments
- Maintaining records, finances, and infrastructure
- Ensuring smooth functioning of the branch

Intern Interaction: While indirect, we benefited from this department's work in ensuring lab availability, internet access, and clean learning spaces.

4. Discussion on Products Studied

During my internship at **Apponix Technologies**, I was exposed to various departments that work cohesively to deliver professional training programs, especially in **Python Full Stack Development**. Below is an overview of the **departments studied**, their roles, and processes followed in the organization.

1. Training and Development Department (Main Focus)

Department Profile:

This department is responsible for designing, developing, and delivering training programs to students. It focuses on professional skill development, particularly in **IT technologies**. The department ensures that the curriculum aligns with current industry trends and equips students with practical skills.

Roles in the Department:

Trainers/Instructors: Lead the sessions, provide theoretical and practical knowledge, and assist with project work.

Course Developers: Work on creating and updating course content to match industry standards.

Support Staff: Assist with student queries, manage class schedules, and ensure smooth operation of training sessions.

Placement Coordinators: Help students find job opportunities post-training by conducting mock interviews, resume building, and liaising with companies.

Process Followed:

Curriculum Design: The training programs are designed based on real-world job requirements. They are constantly updated to include the latest tools and technologies.

Course Delivery: The training is delivered via live sessions (both online and in-person), along with assignments, projects, and assessments.

Practical Exposure: Emphasis is placed on hands-on training, with students working on live projects to apply what they've learned in a real-world setting.

Feedback Mechanism: Regular feedback is collected from students to improve training quality and content delivery.

Communication Type:

Internal Communication: Managed through email, messaging platforms (e.g., Slack), and meetings for coordination among team members.

External Communication: Primarily through email and phone calls with students, placement companies, and other external stakeholders.

2. IT and Infrastructure Department

Department Profile:
Responsible for managing the technical infrastructure needed for delivering training programs, including software, hardware, and network systems.

Roles in the Department:

IT Support Team: Ensures that all systems (computers, servers, etc.) are functioning smoothly and assists with troubleshooting technical issues.

Network Administrators: Maintain the network infrastructure to ensure uninterrupted access to course materials and live sessions.

Process Followed:

System Maintenance: Regular maintenance of the hardware and software used in training and administration.

Security Protocols: Ensuring the security of internal systems, student data, and online course materials.

Student Support: Providing technical support to students experiencing issues with online training platforms.

Communication Type:

Internal Communication: Managed via internal emails and ticketing systems to handle tech-related queries.

External Communication: Communicating with external software providers and service vendors.

3. Placement and Career Development Department

Department Profile:

This department ensures that students are successfully placed after completing their training programs. It includes career counseling, mock interviews, resume building, and liaising with companies for job placements.

Roles in the Department:

Career Counselors: Offer guidance on career paths, job opportunities, and skill requirements.

Placement Coordinators: Work with corporate partners to facilitate job placements and organize campus recruitment drives.

Student Coordinators: Ensure that students are prepared for job interviews and assist with the placement process.

Process Followed:

Resume Building: Helping students craft professional resumes tailored to the IT industry.

Mock Interviews: Conducting mock interviews to prepare students for real-world job interviews.

Industry Partnerships: Collaborating with IT companies to connect students with job opportunities.

Communication Type:

Internal Communication: Conducted through emails, phone calls, and WhatsApp groups for regular updates and announcements.

External Communication: Frequent interaction with corporate clients and recruiters to secure job placements for students.

5. Major Observations During Internship

Throughout my internship, I observed the following key factors that contributed to the success of Apponix Technologies' training programs:

- **Industry-Relevant Curriculum**

  The course content is regularly updated to reflect the latest industry trends, ensuring students are equipped with the most up-to-date knowledge and skills.
- **Practical Training Focus**

  There is a strong emphasis on hands-on projects and real-time scenarios, helping students gain practical experience that is highly valued by employers.
- **Expert Trainers**

  Trainers at Apponix have over six years of professional experience, delivering clear, practical, and industry-aligned content.
- **Placement Assistance**

  The department offers dedicated support for resume building, mock interviews, and job placements, ensuring that students are well-prepared for the job market.
- **Diverse Student Community**

  The training programs attract students from diverse educational and professional backgrounds, fostering collaborative learning and networking opportunities.

6. Overview of Training Received

Duration: 8 Weeks

Location: Apponix Technologies Pvt. Ltd., Hubli Branch

Internship Role: Intern Trainee – Python Full Stack Development

Throughout the internship, we received hands-on training focused on developing practical skills in full-stack web development. The training was structured and delivered by experienced professionals and covered the following key areas:

The training has been a comprehensive journey through multiple stages of learning, covering both fundamental concepts and advanced techniques across Python programming, web development with Django, and integrating APIs. The training included a mixture of theoretical knowledge and practical hands-on experience to enhance my understanding of core programming skills and how they apply to real-world applications.

Python Basics

- **Introduction to Python Syntax**: I began with the basics of Python, learning about its simplicity, readability, and versatility. The training covered fundamental topics like variables, data types, and basic operators, allowing me to start writing simple programs like "Hello, World!".
- **SQL Overview**: Alongside Python basics, I was introduced to SQL, focusing on the structure and use of queries for managing data in relational databases. This gave me a foundational understanding of data storage and retrieval techniques.

Deepening Python Knowledge

- **Operators and Control Structures**: The second week was dedicated to exploring operators (arithmetic, logical, etc.) and control structures (if-else, loops) in Python. This helped in developing problem-solving skills through hands-on coding exercises.
- **Comparison with Other Programming Languages**: I compared Python with other languages like Java and C++, understanding how Python's simplicity and dynamic nature make it ideal for rapid development.
- **Lists, Tuples, and SQL Queries**: Practical experience with Python lists, tuples, and SQL query constraints helped me solidify my understanding of handling data and performing basic database operations.

Advanced Python Concepts and Web Development

- **Functions, Arguments, and Scope**: I learned about Python functions, including defining them, passing arguments, and understanding variable scope. This knowledge helped me structure code more efficiently.
- **String Manipulation**: I explored various string operations in Python, including slicing, indexing, and using methods like strip(), replace(), and formatting techniques for dynamic text.
- **Data Structures**: Detailed lessons on lists, sets, tuples, and dictionaries gave me a deeper understanding of their properties and how they are used in various problem-solving scenarios.
- **File Handling and Exception Management**: I also learned how to work with files in Python, including reading and writing files and managing errors using exception handling.
- **HTML and CSS**: I started exploring web development basics, learning how to create a basic webpage using HTML and CSS, which laid the foundation for integrating Python with web technologies.

Django Basics

- **Django Project Setup**: Week 4 focused on learning Django, starting with setting up a Django project and understanding its directory structure. I created views and worked with URLs to map views to web pages, gaining insight into the basics of web development with Django.
- **Template Handling**: I learned how to use Django's templating engine to create dynamic web pages. This included passing data from views to templates and using template tags for conditional logic and iteration.

Advanced Django Techniques

- **Template Inheritance and URL Routing**: I explored Django's URL routing system and template inheritance, which made it easier to manage reusable components like headers and footers across multiple pages.
- **Bootstrap Integration**: I integrated Bootstrap into Django to make the webpages more responsive and visually appealing.
- **Form Handling in Django**: I created forms in Django using the forms.ModelForm class and learned how to handle user input securely by implementing CSRF tokens.

Advanced Django Concepts and Web Applications

- **HTTP Methods and Database Management**: I deepened my understanding of HTTP methods (POST, GET, PUT, DELETE) and applied them in Django to handle user requests. Additionally, I learned how to use Django's database migration commands to manage schema updates.
- **Session Management**: I explored how cookies are used to manage session data securely, which helped me understand how user data is managed across requests.
- **To-Do List Project**: I worked on building a "To-Do List" application, which involved setting up models, views, and templates to handle CRUD operations and persist data in a database.

API Integration and Data Handling

- **API Integrations in Django**: I learned how to make HTTP requests to external APIs and handle their responses using Python's requests library. I worked with JSON responses and integrated data from APIs into Django templates to display dynamic content on webpages.
- **Open-weather API**: I specifically explored how to integrate the OpenWeather API to fetch weather data and display it on a Django web page. This provided a practical example of how to work with external data and present it to users.

Blog Project Development and HTML5 Integration

- **Project Setup**: I created a new Django project named "Blog" and an app named blog_app, understanding the structure of a Django project.
- **Model Creation**: I defined models like Post and Category to structure blog content and associated the models with the admin interface for easy management.
- **Views and URL Routing**: I created views to display blog posts, post details, and posts by category, while also setting up proper URL routing to handle these views.
- **Base Template Creation**: Designed a base.html template with navigation and content blocks for dynamic rendering, and created various templates for different pages like the post list and category-based posts.
- **HTML5 Features**: I explored advanced HTML5 features such as <canvas>, <video>, and <datalist> to enhance webpage functionality and structure. I also worked with semantic tags to improve the overall page design.

7. Project Assignment Details

During the internship, I was assigned various practical tasks that allowed me to apply the skills I learned. Below are the key projects:

1. Student Registration Form

Objective:

To develop a web-based form for registering students into various courses.

Key Features:

- Input fields for name, email, phone number, course preference, etc.
- Client-side validation using JavaScript to ensure data accuracy.
- Backend integration with MySQL to store student records.
- Confirmation message displayed upon successful registration.

Technologies Used:

HTML, CSS (Bootstrap), JavaScript, Python (Django), MySQL

2. To-Do List Application

Objective:

To create a simple task management system for users to add, view, and manage their daily tasks.

Key Features:

- Add new tasks with description and due date.
- Option to mark tasks as complete or delete them.
- Task storage and retrieval using MySQL database.
- Sorted task display with an intuitive UI.

Technologies Used:

HTML, CSS (Bootstrap), JavaScript, Python (Django), MySQL

3. Login Page

Objective:

To implement a secure login system allowing users to authenticate using credentials.

Key Features:

- Input fields for username and password.
- Client-side input validation.
- User authentication using Django's built-in authentication system.
- Redirects to the user dashboard upon successful login.

Technologies Used:

HTML, CSS (Bootstrap), JavaScript, Python (Django), MySQL

4. Blog Project

Objective:

To develop a blogging platform where users can create, view, and categorize blog posts.

Key Features:

- Post creation with title, content, and category selection.
- Dynamic page rendering using Django templates.
- Integration of categories to filter posts by their respective categories.
- Ability to view detailed blog posts.
- HTML5 integration with <canvas>, <video>, and <datalist> to enhance content interaction.
- Utilized Django's admin interface for managing posts and categories.

Technologies Used:

HTML, CSS (Bootstrap), JavaScript, Python (Django), MySQL

8. Conclusion

The 8-week internship at **Apponix Technologies, Hubli**, was a highly valuable and enriching experience. It provided in-depth, hands-on exposure to key technologies involved in **Python Full Stack Development**, including front-end design, back-end programming with Django, and database management using MySQL.

Working on real-time projects such as the **Student Registration Form**, **To-Do List Application**, and **Login Page** helped bridge the gap between theoretical knowledge and practical implementation. The mentorship and guidance from experienced trainers contributed immensely to my learning curve and boosted my confidence in developing full-stack web applications.

Overall, this internship not only strengthened my technical skills but also introduced me to industry-standard development practices, version control systems like Git, and the importance of clean coding and teamwork—preparing me for future professional opportunities in the software development field.

9. Reference Books, Sites, and Other Resources

Websites:

- https://www.apponix.com/
- https://www.w3schools.com/
- https://docs.djangoproject.com/en/5.2/

Books:

1. *"Automate the Boring Stuff with Python"* by Al Sweigart
   A great resource for Python beginners to automate tasks and become proficient in Python programming.
2. *"Learning Web Design"* by Jennifer Robbins
   Provides foundational knowledge for web design, including HTML, CSS, and JavaScript basics.
3. *"Python Crash Course"* by Eric Matthes
   A comprehensive guide to learning Python through project-based examples, ideal for both beginners and intermediate learners.

Tools Used:

- **Visual Studio Code (VS Code):**
  An integrated development environment (IDE) used for Python development, known for its flexibility and wide range of extensions.
- **SQLiteOnline:**
  An online tool for database management, used for testing MySQL queries and basic database interactions during the internship.

10. Weekly Report

Internship Start Date: 04-12-2024

Week: 1st week

Report Date: From: 04-12-2024 **To**: 06-12-2024

Sl No	Date	Particulars of Work
1	04-12-2024	On the first day, I explored the basics of Python and understood why it is essential in modern programming. Python stands out because of its simplicity, versatility, and extensive libraries, which make it a popular choice for data analysis, web development, automation, and artificial intelligence. Its beginner-friendly syntax allows developers to focus more on problem-solving rather than dealing with complex coding constructs.
2	05-12-2024	Today, I was focused on writing my first Python program, the classic "Hello, World!". This simple exercise helped me understand Python's straightforward syntax and how to execute code using an interpreter or IDE. Alongside this, I learned about the importance of SQL (Structured Query Language) in managing and interacting with databases. SQL is crucial for querying, updating, and organizing data in relational databases, which are foundational for data-driven applications. Understanding SQL will be essential for working with databases effectively in Python projects, especially when dealing with large datasets or web applications.
3	06-12-2024	I dived into declaring variables in Python. Variables are essential for storing and manipulating data, and Python makes this easy with its dynamic typing, allowing variables to change types as needed. Additionally, I began learning about SQL constraints, which are critical for ensuring data integrity in databases. Common constraints like PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, and CHECK help enforce rules for data consistency and reliability, ensuring that the data adheres to specific conditions

		before being stored in the database.

Week: 2nd week

Report Date: **From:**09-12-2024 **To:** 14-12-2024

Sl No	Date	Particulars of Work
1	09/12/2024	**Learning Variables and Operators** I deepened my understanding of variables in Python and explored various types of operators, such as arithmetic, logical, comparison, and assignment operators. These tools are fundamental for performing calculations, making decisions, and evaluating expressions in Python programs. To reinforce the theory, I engaged in hands-on practice by writing programs to perform arithmetic operations and logical comparisons. This practical experience helped solidify my understanding of how operators work and how to use them effectively for problem-solving.
2	10/12/2024	**Python vs. Other Programming Languages** The day was dedicated to comparing Python with other programming languages. I learned how Python's simplicity and readability set it apart from more complex languages like Java or C++. Python's dynamic typing and concise syntax make it easier to learn and implement compared to the stricter, more verbose syntax of other languages. I also revisited my first Python program, "Hello, World!", which further reinforced why Python is widely adopted across domains like web development, data science, and machine learning. Its ease of use ensures quicker development and a reduced learning curve, making it a preferred choice for many developers.
3	11/12/2024	**Focused on Python's Control Structures** I focused on Python's control structures, which are essential for making decisions and iterating through code. I learned about conditional statements (if, else, elif) and loops (for, while). To practice these concepts, I wrote programs to compute factorials and Fibonacci sequences. Additionally, I implemented a program where a user was repeatedly prompted to input the

		correct password until they got it right, using loops without predefined functions. These exercises highlighted the power and flexibility of control structures in Python and their crucial role in real-world scenarios, where dynamic decision-making and iteration are often required..
3	12/12/2024	**Lists, SQL Queries, and DBMS Comparison** The fourth day was a blend of Python and database concepts. I explored Python lists, learning how to create, access, and manipulate list elements. At the same time, I delved into SQL queries and gained an understanding of the differences between relational and non-relational database management systems (DBMS). Relational DBMS (RDBMS) like MySQL and PostgreSQL store data in structured tables, while non-relational DBMS like MongoDB handle unstructured or semi-structured data in collections. To support my learning, I installed relevant software such as MySQL Workbench and Python IDEs, enabling me to practice both database interactions and Python programming seamlessly.
4	13/12/2024	**Tuples and Database Creation** I learned about tuples in Python, which are immutable sequences. Their immutability makes them ideal for storing fixed data that should not be changed. On the database side, I created a college database, defined a students table, and inserted some records. Using SQL queries, I retrieved and manipulated the data, which helped solidify my understanding of database operations. The integration of Python and SQL further demonstrated how these technologies complement each other in real-world applications, offering powerful solutions for data storage and management.
5	14/12/2024	**Revisiting and Practicing Concepts** The day was dedicated to revisiting and practicing all the topics I had learned so far. Since there was no office, I utilized the time at home to strengthen my knowledge of Python variables,

		control structures, lists, tuples, and SQL concepts. I spent the day writing programs and executing SQL queries, which helped me gain more confidence in applying these concepts effectively and deepened my understanding of how they work together in real-world scenarios.

Week: 3rd week

Report Date: **From:** 16-12-2024 **To:** 21-12-2024

Sl No	Date	Particulars of Work
1	16/12/2024	**Understanding Functions in Python** I focused on understanding the concept of functions in Python, which are essential for writing reusable and modular code. I learned how to define functions using the def keyword and practiced writing user-defined functions with and without parameters. I also explored built-in Python functions like len(), type(), and print(), which helped me understand how functions can simplify coding tasks. Additionally, I implemented return statements in functions to fetch results and pass data between functions. Hands-on exercises allowed me to apply these concepts in practical scenarios, such as creating functions for basic mathematical operations (addition, subtraction) and more complex operations like calculating the factorial of a number.
2	17/12/2024	**Exploring Function Arguments and Variable Scope** On Day 2, I delved deeper into function arguments. I studied different types of arguments, including positional, keyword, and default arguments, and explored how arbitrary arguments (e.g., *args and **kwargs) can handle varying numbers of input parameters. This flexibility in function arguments helped me understand how to create more dynamic and versatile functions. Additionally, I learned about variable scope—how variables can be local, global, or nonlocal—and how these scopes affect variable access inside and outside functions. I practiced defining functions that utilized different scopes and used the global keyword to modify global variables from within functions. This was crucial for understanding how data flows and is manipulated within Python programs.
3	18/12/2024	**String Manipulation in Python** I focused on string manipulation, an essential skill in Python. I learned how to perform basic string operations, such as slicing

		and indexing, and explored string methods like lower(), upper(), strip(), split(), replace(), and find(). These methods helped me clean, format, and search strings more efficiently. I also explored string formatting techniques using f-strings and the format() method, which made it easier to insert dynamic data into strings. To solidify these concepts, I worked on hands-on exercises, such as checking if a string is a palindrome and performing text analysis to count the number of vowels and words in a given sentence.
13	19/12/2024	**Lists, Tuples, and Sets in Detail** I focused on understanding different data structures—lists, tuples, and sets. I revisited lists and studied list comprehensions, which provide a concise way to create and manipulate lists. I also learned about tuples, which are immutable sequences, and explored how they differ from lists in terms of mutability and use cases. Sets were another important topic I explored, particularly their properties and operations like union, intersection, and difference. To practice these concepts, I solved problems using these data structures, such as removing duplicates from a list and finding common elements between two sets. By manipulating these structures, I gained a deeper understanding of their unique properties and how they can be effectively used in Python programs.
4	20/12/2024	**Dictionaries and File Handling** I focused on a blend of learning about dictionaries and handling files. I learned how dictionaries store key-value pairs and how to use methods like get(), keys(), values(), and items() to access and manipulate data. I also worked on creating nested dictionaries and explored how to access deeply nested elements. The second part of the day was dedicated to file handling, where I learned how to open, read, write, and append files using various modes (r, w, a, r+). I practiced writing Python programs to process text files, count the number of words, and store the results in new files. Additionally, I learned

		how to manage errors during file operations using exception handling (try-except), which is crucial for making code more robust.
5	21/12/2024	**Introduction to HTML, CSS, and JavaScript** I was introduced to the basics of web development with HTML, CSS, and JavaScript. I began by studying the structure of an HTML document and familiarized myself with essential tags like <h1>, <p>, <a>, <img>, and forms (<form>, <input>, <button>). I learned how to create a basic webpage with headings, paragraphs, and links, and then styled it using CSS. I explored CSS selectors, including tag, class, and ID selectors, and learned how to apply various styling properties like color, font-size, margin, padding, and background-color. By understanding the Box Model, I was able to create layouts that properly manage spacing and element positioning. To enhance the webpage's appearance, I integrated CSS with the HTML structure. I also explored the basics of JavaScript, including how to add interactivity to a webpage by manipulating HTML elements, handling user events, and using basic functions like alert() and console.log(). Additionally, I learned how Python frameworks like Flask and Django can integrate with web technologies to build dynamic web applications, which prepared me for combining Python with HTML, CSS, and JavaScript in future projects.

Week: 4th week

Report Date: **From:** 23-12-2024 **To:** 27-12-2024

Sl No	Date	Particulars of Work
1	23/12/2024	**Diving into Django's Core Features** I dove into Django's core features, focusing on the creation and structure of Django projects. I learned how to set up a Django project and app, exploring the directory structure that Django automatically sets up. I familiarized myself with the settings file, where important configurations are stored, and how Django uses URLs to map views to web pages. I also worked on creating my first view in Django, which returned a simple response to the browser. This was an exciting step as it helped me understand the foundational components of working with Django.
2	24/12/2024	**Focusing on Django Views and Templates** I focused on Django views and templates. I explored how views handle HTTP requests and return HTTP responses. I learned how to create different views to handle various types of requests, such as GET and POST, and how these views map to specific URLs in the project. I also began learning about Django's templating engine, which allows dynamic content to be displayed on web pages. I practiced passing data from views to templates and rendering HTML content dynamically, which helped me understand how Django generates dynamic web pages.
3	25/12/2024	**Understanding Django's URL Routing System** I focused on understanding Django's URL routing system. I learned how to use the urls.py file to map URLs to views, enabling navigation within the application. I practiced creating different routes for various pages and learned how to pass parameters through the URL. I also explored how to organize the URL configuration into reusable patterns, making the

		project more scalable. This helped me understand how Django handles routing and how URLs are structured in Django applications.
4	26/12/2024	**Work on Django Views and Templates** I focused on using Django template tags and filters to manipulate and display data dynamically in HTML templates. I practiced using template tags like {% if %}, {% for %}, and {% block %}, which allowed me to add logic to my templates. I also learned about inheritance in templates, enabling me to create a base template with common elements (like headers and footers) and extend it in other templates. This approach helped me understand how Django promotes code reuse and efficient organization.
5	27/12/2024	**Continuing Django Development and Exploring Advanced Features** I continued to build my Django skills by reviewing everything I had learned. I integrated views, templates, and URL routing into a small, functional project. I also explored more advanced features like form handling in Django, learning how to create forms and handle user input. This step helped me understand how Django facilitates the creation of interactive and user-friendly web applications. It also highlighted how to effectively manage both static and dynamic content in a web application.

Week: 5th week

Report Date: From: 30-12-2024 **To:** 03-01-2025

Sl No	Date	Particulars of Work
1	30/12/2024	**Creating a Django Project with an "About Us" Page** I created a Django project and developed a mini application within it, starting with an "About Us" page. I explored how to integrate Bootstrap into my Django application to enhance the design and responsiveness of the page. Additionally, I learned about Django Template Language (DTL), specifically how to use the {% if %} tag to add conditional logic in templates. This enabled me to display dynamic content based on certain conditions, making the page more interactive.
2	31/12/2024	**Learning Django URL Routing and Template Data Handling** I focused on learning how to link URLs in Django using their names. By using Django's URL routing system, I linked different pages within the application by referencing their names in the urls.py file. This approach made the code cleaner and more maintainable. I also worked further with Django Template Language (DTL), learning how to pass context data from views to templates and render dynamic content on web pages. This deepened my understanding of how data flows from the backend (views) to the frontend (templates) in Django.
3	01/01/2025	Holliday.
4	02/01/2025	**Improving Design with Bootstrap in Django** I focused on improving the design of my Django project by integrating more Bootstrap components. I created a "Contact" page and worked on refining the layout of my mini website. The use of Bootstrap significantly enhanced the visual appeal of the site, making it more user-friendly and responsive. This was a key step in making my website more interactive and polished.
5	03/01/2025	**Working with Django's Template Inheritance System**

		I worked with Django's template inheritance system and explored how to use the {% extends %} and {% block %} tags. This allowed me to reuse HTML code across multiple pages. By creating a base template, I could extend it in other templates and define specific content for each page within blocks. This approach made my code more modular and easier to maintain, as common elements like headers and footers were reusable across all pages of the website.

Week: 6th week

Report Date: **From:** 06-01-2025 **To**: 11-01-2025

Sl No	Date	Particulars of Work
1	06/01/2025	**Learning Django Forms and Security** I started the week by learning how to create forms in Django using the forms.ModelForm class, which simplifies form creation by utilizing model attributes. I implemented a login form with fields for username and password and used the {{ form.as_p }} syntax to render the form fields as paragraphs, improving readability. Additionally, I added a CSRF token to the form to enhance security by preventing cross-site request forgery (CSRF) attacks. During this session, I also explored how cookies are used to manage session data securely and how the CSRF token uses cookies for validation.
2	07/01/2025	**Exploring HTTP Methods and Django Commands** Continuing with the login form, I delved into HTTP methods, including POST, GET, PUT, and DELETE. I gained a solid understanding of how each method is used in web development. I also discussed the differences between PATCH and PUT: PATCH is used for partial updates, while PUT is intended for full updates of resources. To further enhance my knowledge, I practiced using the Django commands py manage.py migrate and py manage.py makemigrations, which helped me apply and manage database changes effectively.
3	08/01/2025	**Review and Clarification of Concepts** I revisited the concepts covered on Monday and Tuesday to reinforce my understanding. During this review, I clarified doubts about Django stacks, particularly focusing on the flow of HTTP methods and how they integrate with Django's request-response cycle. This session helped solidify my knowledge and boosted my confidence in working with Django.
4	09/01/2025	**Holiday due to Protests in Hubli**

		This was a holiday due to protests in Hubli. I used this time to reflect on the concepts learned so far and plan the upcoming tasks for the week.
5	10/01/2025	I started a new project, a "To-Do List" application. My work involved designing the structure of the application and creating the necessary modules and views for managing tasks. I set up models to define the task attributes, and developed views to handle various task operations such as adding, updating, and deleting tasks. Additionally, I planned the logic required for these functionalities to ensure smooth task management within the application.
6	11/01/2025	I integrated the "To-Do List" application with a database to store and manage tasks persistently. I enhanced the user interface using Bootstrap, ensuring the design was visually appealing and user-friendly. Additionally, I tested the CRUD (Create, Read, Update, Delete) operations to ensure the application functioned smoothly and efficiently.

Week: 7th week

Report Date: **From:** 13-01-2025 **To**: 17-01-2025

Sl No	Date	Particulars of Work
1	13/01/2025	I began the week by learning about API integrations in Django, focusing on how to make HTTP requests to external APIs and process their responses. I explored the use of Python libraries like requests to interact with APIs and discussed best practices for managing API keys securely. Additionally, I learned how to handle JSON responses and display the fetched data dynamically on Django templates.
2	14/01/2025	Holiday for Makar Sankranti.
3	15/01/2025	I reviewed the migration process in Django to ensure smooth database schema updates. This included the use of makemigrations and migrate commands, along with a discussion on how to resolve common issues such as conflicts or missing migrations. I also revisited the previous day's API concepts and clarified doubts about error handling in API calls
4	16/01/2025	I learned about the OpenWeather API and its integration into Django projects. I began by signing up for an account and obtaining an API key, which was necessary for authenticating my requests. After integrating the API into my Django project, I used Python's requests library to send GET requests to fetch real-time weather data for a specific city. The API returned JSON responses containing key weather details such as temperature, humidity, wind speed, and weather conditions. I then parsed this data and explored how to dynamically present this information on a Django web page
5	17/01/2025	I focused on connecting API data with Django views and templates. This involved passing the data retrieved from APIs to Django templates using context dictionaries, allowing the dynamic display of real-time information. I learned how to structure the data in a way that could be easily accessed within templates. By using Django Template Language (DTL), I was

		able to render this data, such as weather details, in a user-friendly format. I made sure the information updated correctly in response to API calls, ensuring that users always received accurate and up-to-date content when interacting with the web interface.

Week: 8th week

Report Date: From: 20-01-2025 **To**: 25-01-2025

Sl No	Date	Particulars of Work
1	20/01/2025	**Project Setup:** I created a new Django project named "Blog" using django-admin startproject blog. I ensured the project directory was properly structured, set up a virtual environment, and installed Django to manage dependencies cleanly. **App Creation:** I created a new app named blog_app using python manage.py startapp blog_app. I then added blog_app to the INSTALLED_APPS list in the settings.py file. Finally, I ran the server using python manage.py runserver to confirm the project was working correctly.
2	21/01/2025	**Model Creation:** In blog_app/models.py, I defined the Post model with fields: title (char field), content (text field), author (char field), created_at and updated_at (datetime fields). I also created a Category model with fields: name (char field) and slug (slug field). These fields were chosen to support post categorization and detailed blog content. **Migration and Registration:** I added __str__ methods for both models to display their names clearly in the admin interface. Then, I created and applied migrations using python manage.py makemigrations and python manage.py migrate. After that, I registered these models in admin.py to manage them from the Django admin panel.
3	22/01/2025	**View Functions:** I created three views in blog_app/views.py: post_list for displaying all posts, post_detail for showing the details of a single post, and category_posts for filtering posts by category. Each view queried the database using the respective model and returned an HTTP response. **URL Routing:** I set up URL patterns in blog_app/urls.py for the views created earlier, associating paths like /posts/,

		/post/<int:id>/, and /category/<slug:slug>/ with their respective views. I then included these patterns in the project-level urls.py using the include() function to ensure proper routing across the app.
4	23/01/2025	**Base Template Creation:** I created a templates/blog_app directory and designed a base template named base.html. It included a navigation bar for links like "Home," "Categories," and "About," as well as a {% block content %} for dynamically rendering content in child templates. **Dynamic Templates:** I built three templates: post_list.html to display a list of blog posts, post_detail.html to show detailed information about a single post, and category_posts.html to show posts filtered by category. I used Django Template Language (DTL) tags like {% for %} and {% if %} to iterate over data and display content dynamically.
5	24/01/2025	**Adding Dummy Data:** Using the Django admin panel, I added dummy categories and posts with realistic titles, content, and timestamps to populate the database for testing purposes. This allowed me to verify that the models were functioning as intended. **Template Integration:** I updated the views to query the database and pass the fetched data as context variables to the templates. For instance, in post_list.html, I displayed all posts using a loop and linked each post to its detail page. I tested the functionality by navigating through pages to ensure proper display and routing.
6	25/01/2025	Add a "Categories" page that lists all categories and links to their respective filtered posts. Update the post_list view to include the created_at date under each post title. I explored **HTML5 advanced features**, including <canvas>

		for creating simple graphics, <video> for embedding multimedia, and <datalist> for enhanced form inputs. I also learned about semantic tags like <article> and <section> for better webpage structure.

Week: 9th week

Report Date: **From:** 27-01-2025 **To**: 29-01-2025

1	27/01/2025	Add a "Contact" page with a simple form containing fields for name, email, and message (no backend functionality yet). Update the navigation menu to include links to the "Contact" and "About" pages. I focused on **CSS advanced topics**, including animations using @keyframes, transitions for smooth effects, and flexbox for building responsive layouts. I also experimented with media queries for designing mobile-friendly pages.
2	28/01/2025	Add a "Contact" page with a simple form containing fields for name, email, and message (no backend functionality yet). Update the navigation menu to include links to the "Contact" and "About" pages. I focused on **CSS advanced topics**, including animations using @keyframes, transitions for smooth effects, and flexbox for building responsive layouts. I also experimented with media queries for designing mobile-friendly pages.
3	29/01/2025	**Add a search bar to the post_list.html template to filter posts by title. Update the post_list view to handle search queries.** Test all features, ensuring proper navigation and functionality across the app. I focused on **JavaScript DOM manipulation**, including modifying elements, handling events, and dynamically creating content. I also explored localStorage and sessionStorage for storing data on the client side.