



PURNEA COLLEGE OF ENGINEERING, PURNEA

An Internship Report on **PROGRAMMING WITH PYTHON** UNDERTAKEN **INTERNBEE TRAININGS**

Submitted in Partial fulfilment of the requirements for the award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (2k21 – 2k25)

Submitted to :

Prof. RAJU KUMAR

(H.O.D of C.S.E)

Submitted by :

Name : ASHUTOSH KUMAR

Reg. No. : 21105131015

Semester : 7th

Internal Examiner :

External Examiner :

Ashutosh

06 Dec 2024 – 31 Jan 2025

Acknowledgment & Executive Summary

Acknowledgment

I express my deepest gratitude to INTERNBEE Trainings for providing me with the opportunity to intern at their organization. I am especially thankful to my colleagues and mentors for their continuous guidance and support throughout the internship. Their experience and insights have been invaluable to my learning process and professional development.

Executive Summary

This report provides an overall description of my internship experience at INTERNBEE Trainings from December 6, 2024, to January 31, 2025. The internship focused on learning and applying Python programming, covering topics ranging from core syntax to real-world applications. This document outlines my weekly progress, challenges encountered, skills acquired, and the overall impact of the internship on both my technical expertise and professional growth.

Table of Contents

Section No.	Section Title	Page No.
1	Introduction	3
2	Company Overview	4
3	Training Modules	5-6
4	Weekly Progress	-----
4.1	Week 1 (Module 1 & 2)	7
4.2	Week 2 (Module 3)	7-8
4.3	Week 3 (Module 4)	8
4.4	Week 4 (Module 5)	8-9
4.5	Week 5 (Module 6 & 7)	9
4.6	Week 6 (Module 8)	9-10
4.7	Week 7 (Module 9)	10
4.8	Week 8 (Module 10)	10
5	Certificate of Completion	11
6	Skills & Learning Outcomes	12
7	Challenges & Solutions	13
8	Conclusion & Recommendations	14
9	References	15
10	Appendices	
10.1	Appendix A	16-17
10.2	Appendix B	17-18
10.3	Appendix C	18-20

1. Introduction

Purpose of Internship

The goal of this internship was to gain in-depth programming experience in Python and apply theoretical knowledge in practical contexts. It aimed to bridge the gap between academic learning and real-world implementation.

Internship Objectives

- Master Python programming.
- Develop problem-solving and debugging skills.
- Explore Python applications across various domains.
- Work on real-world projects to gain advanced technical expertise.

Responsibilities & Role

As an intern, my primary responsibilities included working through assigned Python modules, applying learned concepts through coding exercises, and actively participating in problem-solving sessions. I was expected to demonstrate logical reasoning, optimize coding efficiency, and document key takeaways from my learning experience.

2. Company Overview

Company History, Mission, and Vision

INTERNBEE Trainings is an organization with a strong commitment to providing skill-based training programs for both students and professionals. Their vision is to empower individuals with advanced technological knowledge to enhance their career growth. Their mission is to bridge the skills gap in the industry by offering hands-on learning experiences aligned with market demands.

Organizational Structure

The organization follows a structured system comprising training instructors, mentors, and support staff, all working together to ensure the smooth execution of training programs.

Industry Overview

The education and training industry is evolving rapidly, with a growing demand for online learning and skill development courses. Organizations like INTERNBEE Trainings play a crucial role in equipping learners with the essential technical expertise required to meet industry standards.

3. Training Modules

Module 1: Introduction to Python

- Overview of Python and its features
- Setting up the Python environment (interpreter, IDEs)
- Basic syntax, data types, variables, and operators

Module 2: Control Flow and Functions

- Control structures: loops and conditional statements
- Function declaration, invocation, and parameters
- Scope and lifetime of variables

Module 3: Data Structures in Python

- Lists, tuples, sets, and dictionaries
- Manipulating and iterating through data structures
- List comprehensions and built-in functions

Module 4: File Handling and I/O Operations

- Reading from and writing to files in Python
- Working with different file formats (CSV, JSON, etc.)
- Error handling and exception mechanisms

Module 5: Object-Oriented Programming (OOP) in Python

- OOP concepts: classes, objects, inheritance, and polymorphism
- Encapsulation, abstraction, and class methods
- Special methods (dunder methods) and operator overloading

Module 6: Modules, Packages, and Libraries

- Working with modules and importing functionality
- Creating and using Python packages
- Introduction to popular libraries (NumPy, Pandas, Matplotlib)

Module 7: Working with Data and Data Manipulation

- Data analysis and manipulation using Pandas
- Data visualization with Matplotlib and Seaborn
- Introduction to data cleaning and preprocessing

Module 8: Introduction to Web Development with Python

- Basics of web development using frameworks like Flask or Django
- Creating web applications and APIs
- Deploying a simple web application

Module 9: Python for Automation and Scripting

- Automating tasks with Python (file manipulation, web scraping)
- Scripting and writing efficient code for various tasks
- Working with regular expressions

Module 10: Real-world Projects and Applications

- Applying Python skills to real-world projects
- Developing small-scale applications or scripts
- Showcasing learned concepts through projects

4. Weekly Progress

4.1 : Week 1

- **Tasks Assigned:**

I was responsible for learning the fundamentals of Python, which included setting up the programming environment, understanding basic syntax, and working with control structures such as loops and functions. Additionally, I explored different IDEs like **Jupyter Notebook** and **VS Code** to enhance my coding experience.

- **Tools & Software Used:**

The primary tools utilized during this phase were **Python**, **Jupyter Notebook**, and **VS Code**, which helped in writing, testing, and debugging Python scripts efficiently.

- **Key Learnings:**

This week provided a strong foundation in Python programming, focusing on understanding basic syntax, implementing loops and conditional statements, and defining and calling functions. Additionally, I gained insights into configuring Python environments for effective development.

- **Challenges & Solutions:**

One of the primary challenges I faced was understanding the **scope and lifetime of variables**, which resulted in unexpected program behaviors. To overcome this, I referred to Python's official documentation, conducted debugging exercises, and practiced writing different functions to observe variable behavior in various scopes. This helped me develop a clearer understanding of the topic and improved my problem-solving skills.

4.2 : Week 2

- **Tasks Assigned:**

This week focused on understanding and working with **data structures in Python**. I explored different built-in data structures, including **lists, tuples, sets, and dictionaries**, and learned how to manipulate them efficiently using Python's built-in functions and methods. Additionally, I practiced using **list comprehensions** to create concise and efficient code.

- **Tools & Software Used:**

The primary tools utilized were **Python** and **Pandas**, which provided powerful functionalities for data manipulation and handling structured data efficiently.

- **Key Learnings:**

This module enhanced my understanding of fundamental data structures and their practical applications in Python. I gained proficiency in creating, modifying, and iterating over **lists, tuples, sets, and dictionaries**, as well as leveraging **list comprehensions** for improved performance and readability.

- **Challenges & Solutions:**

Initially, I encountered difficulties in working with **dictionary operations**, particularly when handling nested dictionaries and retrieving values dynamically. To overcome this, I practiced multiple exercises, explored Python's official documentation, and implemented real-world examples to strengthen my understanding. Through hands-on problem-solving, I was able to grasp the concepts effectively and improve my ability to manipulate complex data structures.

4.3 : Week 3

- **Tasks Assigned:** File handling and I/O operations.
- **Tools & Software Used:** Python, CSV, JSON.
- **Key Learnings:** Reading/writing files, handling exceptions.
- **Challenges & Solutions:** Faced encoding issues while handling text files; resolved using proper encoding formats.

4.4 : Week 4

- **Tasks Assigned:**

This week focused on **Object-Oriented Programming (OOP) concepts** in Python. I worked on implementing **classes, objects, inheritance, polymorphism, and encapsulation**, gaining a deeper understanding of how OOP principles structure and organize code.

- **Tools & Software Used:**

The key tools used were **Python** and OOP methodologies, which helped in designing reusable and modular code.

- **Key Learnings:**

I learned how to define **classes and objects**, implement **inheritance** for code

reusability, use **polymorphism** to enhance flexibility, and apply **encapsulation** to protect data. These concepts improved my ability to write well-structured and efficient programs.

- **Challenges & Solutions:**

Initially, understanding **abstraction** was challenging as it required a conceptual approach rather than direct coding. I overcame this by working on small practice exercises and seeking guidance from mentors, which helped clarify its implementation and real-world applications.

4.5 : Week 5

- **Tasks Assigned:**

This week involved **working with Python modules and packages**, as well as **data manipulation and visualization** using popular libraries.

- **Tools & Software Used:**

Python, NumPy, Pandas, Matplotlib, Seaborn for data handling and visualization.

- **Key Learnings:**

Learned to **import and use modules**, process large datasets efficiently with **Pandas**, perform numerical computations with **NumPy**, and create data visualizations using **Matplotlib and Seaborn**.

- **Challenges & Solutions:**

Faced difficulties in **handling large datasets**, leading to slow performance. Optimized code by using **efficient data structures and vectorized operations**, improving processing speed.

4.6: Week 6

- **Tasks Assigned:**

Focused on **introduction to web development with Python**, including setting up frameworks and building basic applications.

- **Tools & Software Used:**

Worked with **Flask** and **Django** for backend web development.

- **Key Learnings:**

Gained experience in **creating web applications**, integrating **APIs**, and understanding the **deployment process** for web projects.

- **Challenges & Solutions:**

Faced initial difficulties in **configuring the web frameworks**. Overcame them by referring to **official documentation** and following **online tutorials** for proper setup and implementation.

4.7 : Week 7

- **Tasks Assigned:**

Worked on **Python for automation and scripting**, focusing on automating repetitive tasks and extracting data from websites.

- **Tools & Software Used:**

Utilized **Python, web scraping tools (BeautifulSoup, Selenium), and regular expressions** for text processing.

- **Key Learnings:**

Learned to **write automation scripts**, scrape and process **web data**, and use **regular expressions** for pattern matching and data extraction.

- **Challenges & Solutions:**

Faced difficulties in **handling dynamic web elements**. Overcame this by learning to use **Selenium** for interacting with web pages and **BeautifulSoup** for parsing static content efficiently.

4.8 : Week 8 (Project)

- **Project Title : "Live Data Dashboard with CSV Upload"**

- **Tools & Software Used:** Python, Flask, Pandas, Matplotlib, Seaborn, HTML/CSS.

- **Key Learnings:** Implemented **file upload functionality**, processed and cleaned data using **Pandas**, and generated **dynamic visualizations** using **Matplotlib and Seaborn**. Gained hands-on experience in **Flask web development** and user interaction handling.

- **Challenges & Solutions:** Initially faced difficulties in **integrating Flask with Pandas for real-time data updates**. Overcame this by structuring the project efficiently, following best practices for Flask routes, and ensuring seamless front-end integration using templates.

5. Certification of Completion



Certificate Credentials :

Name : Ashutosh Kumar

Date of Certification : 04/02/2025

Certificate Number : 20250205IB1268892058

6. Skills & Learning Outcomes

Technical Skills Learned:

- Proficiency in Python programming.
- Hands-on experience in data handling and manipulation.
- Fundamental knowledge of web development.

Advancements Over Time:

- Improved efficiency in writing optimized and structured code.
- Enhanced debugging skills for identifying and resolving errors effectively.
- Deeper understanding of how programming concepts apply to real-world scenarios.

7. Challenges & Solutions

Key Difficulties Faced:

One of the main challenges I encountered was understanding object-oriented programming (OOP) concepts, including classes, inheritance, and polymorphism. Applying these concepts effectively in Python was initially difficult. Additionally, handling large datasets was challenging, particularly in terms of data cleaning, processing efficiency, and memory management.

Strategies Used to Overcome Challenges:

To overcome these difficulties, I practiced coding exercises regularly and worked on small projects to reinforce OOP concepts. Referring to official Python documentation and online tutorials helped clarify complex topics. For handling large datasets, I explored efficient data manipulation techniques using Pandas and NumPy, focusing on performance optimization. Engaging in discussions with mentors and peers also provided valuable insights and alternative problem-solving approaches.

8. Conclusion & Recommendations

Overall Impact of the Internship:

The internship had a profound impact on my technical and professional growth. It not only strengthened my Python programming skills but also enhanced my problem-solving abilities and logical thinking. Working on real-world applications allowed me to understand how programming concepts are implemented in practical scenarios. Additionally, the structured learning approach and mentorship provided valuable guidance, making the experience both educational and rewarding.

Future Workflow Improvements:

To further enhance the learning experience, I recommend incorporating more hands-on projects that simulate real-world challenges. Live coding sessions with mentors could help clarify complex concepts in real-time, and collaborative assignments would encourage teamwork and peer learning. Additionally, integrating code reviews and feedback sessions could help interns refine their coding practices and adopt industry best practices more effectively.

9. References

1. Python Documentation: *Python Software Foundation*. (2025). *Python 3 Documentation*. Retrieved from <https://docs.python.org/3/>
2. Flask Documentation: *Pallets Projects*. (2025). *Flask Web Framework*. Retrieved from <https://flask.palletsprojects.com/>
3. Pandas Documentation: *Pandas Development Team*. (2025). *Pandas User Guide*. Retrieved from <https://pandas.pydata.org/>
4. Hunter, J. D. (2025). *Matplotlib: Python Plotting Library*. Retrieved from <https://matplotlib.org/stable/>
5. Waskom, M. (2025). *Seaborn: Statistical Data Visualization*. Retrieved from <https://seaborn.pydata.org/>
6. Harris, C. R., et al. (2025). *NumPy User Guide*. Retrieved from <https://numpy.org/doc/stable/>
7. Richardson, L. (2025). *Beautiful Soup Documentation*. Retrieved from <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

10. Appendices

Appendix A : Code snippets

```
app.py > ...
1   from flask import Flask, render_template, request
2   import pandas as pd
3   import os
4   import matplotlib.pyplot as plt
5   import seaborn as sns
6
7   app = Flask(__name__)
8   UPLOAD_FOLDER = "uploads"
9   os.makedirs(UPLOAD_FOLDER, exist_ok=True)
10  app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
11
12 def process_csv(file_path):
13     df = pd.read_csv(file_path)
14     return df
15
16 def generate_plots(df):
17     plot_paths = []
18     numeric_columns = df.select_dtypes(include=['number']).columns
19     if len(numeric_columns) >= 2:
20         plt.figure(figsize=(6, 4))
21         sns.histplot(df[numeric_columns[1]], kde=True, bins=20, color='#007bff')
22         plt.xlabel("Values", fontsize=12, color="#333")
23         plt.ylabel("Frequency", fontsize=12, color="#333")
```

app.py (Flask App)

```
templates > ◊ dashboard.html > ↗ html
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <title>Data Dashboard</title>
5       <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
6   <style>
7       body {
8           background: linear-gradient(135deg, #007bff, #6610f2);
9           padding: 20px;
10          color: white;
11          font-family: 'Arial', sans-serif;
12      }
```

Dashboard.html (dashboard page)

```

templates > upload.html > html > head > style > .btn:hover
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <title>Upload File</title>
5       <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
6   <style>
7       body {
8           background: linear-gradient(135deg, #007bff, #6610f2);
9           text-align: center;
10          padding: 50px;
11          color: white;
12          font-family: 'Arial', sans-serif;
13      }
14     .container {
15         max-width: 500px;
16         background: white;
17         padding: 30px;
18         border-radius: 15px;
19         box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.2);
20         animation: fadeIn 1s ease-in-out;
21     }

```

upload.html (Upload page)

Appendix B : Terminal snippets

```

PS C:\Users\asus\OneDrive\Desktop\internbee_proj> pip install flask pandas matplotlib seaborn
DEPRECATION: Loading egg at c:\python311\lib\site-packages\vboxapi-1.0-py3.11.egg is deprecated.
pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330
Requirement already satisfied: flask in c:\python311\lib\site-packages (2.2.2)
Requirement already satisfied: pandas in c:\python311\lib\site-packages (2.1.4)
Requirement already satisfied: matplotlib in c:\python311\lib\site-packages (3.8.2)
Requirement already satisfied: seaborn in c:\python311\lib\site-packages (0.13.1)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\python311\lib\site-packages (from flask) (2.2.2)
Requirement already satisfied: Jinja2>=3.0 in c:\python311\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\python311\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.0 in c:\python311\lib\site-packages (from flask) (8.1.3)
Requirement already satisfied: numpy<2,>=1.23.2 in c:\python311\lib\site-packages (from pandas) (1.26.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\asus\appdata\roaming\python\python311\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\python311\lib\site-packages (from pandas) (2023.3.post1)

```

Installing Dependencies

```
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\roaming\py  
ckages (from python-dateutil>=2.8.2->pandas) (1.16.0)  
  
[notice] A new release of pip is available: 23.3.2 -> 25.0.1  
[notice] To update, run: python.exe -m pip install --upgrade pip  
PS C:\Users\asus\OneDrive\Desktop\internbee_proj>  
PS C:\Users\asus\OneDrive\Desktop\internbee_proj> python app.py
```

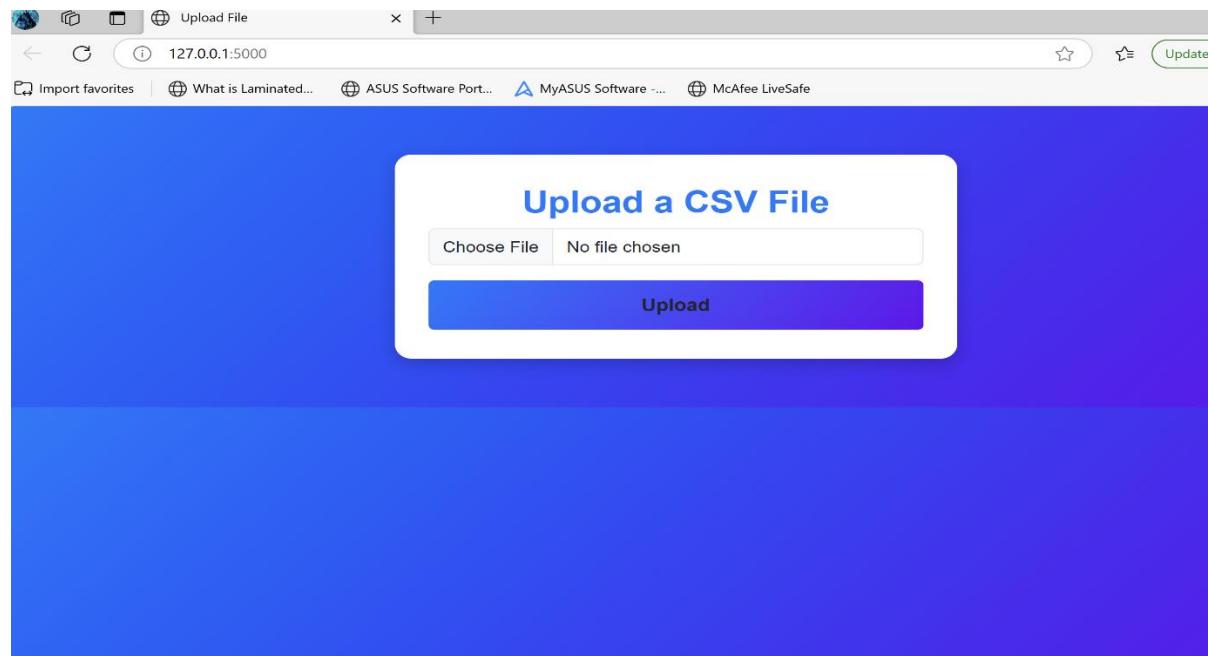
Running Flask Application

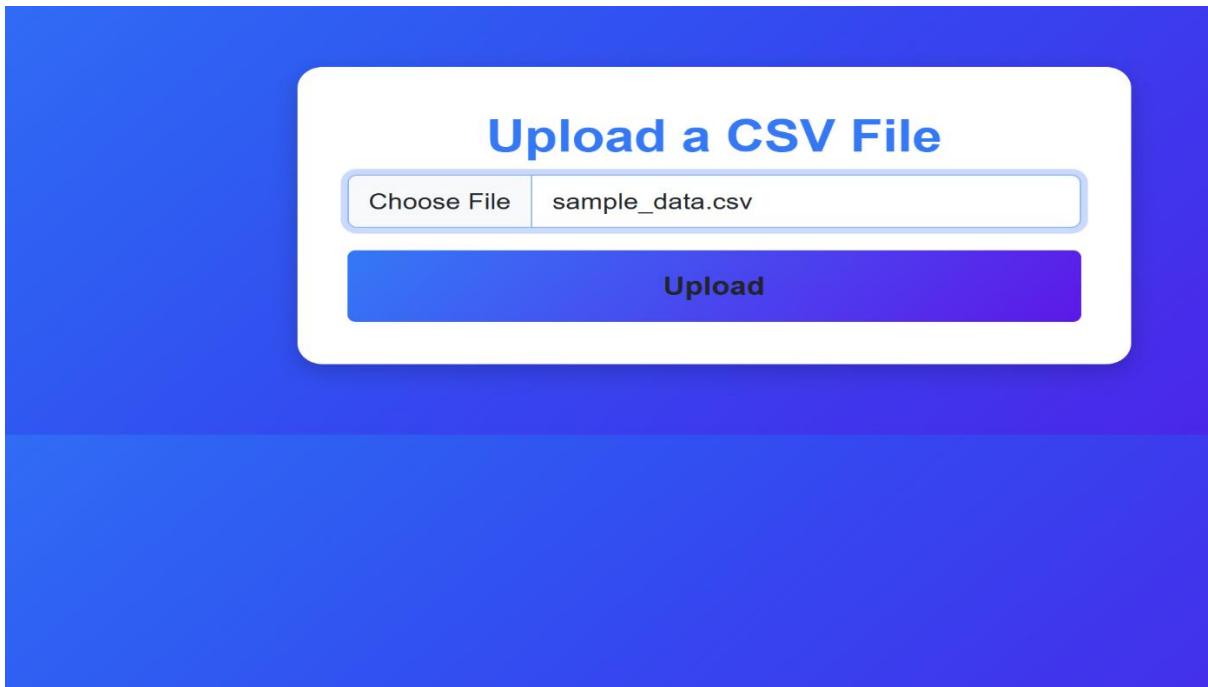
CMD : python app.py

```
Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production  
WSGI server instead.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 697-082-936
```

Running the app on local host port number 5000

Appendix C : Interface Captures & Rendered Charts





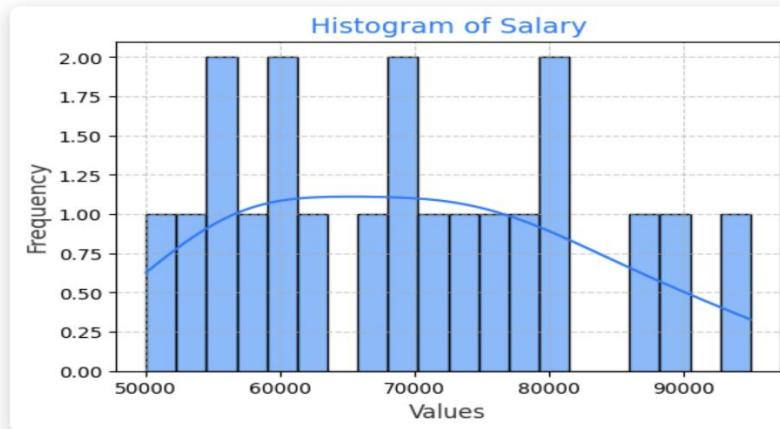
Upload page

A screenshot of a browser window titled "Data Dashboard" at address "127.0.0.1:5000". The main content area displays a table titled "Uploaded Data" with the following data:

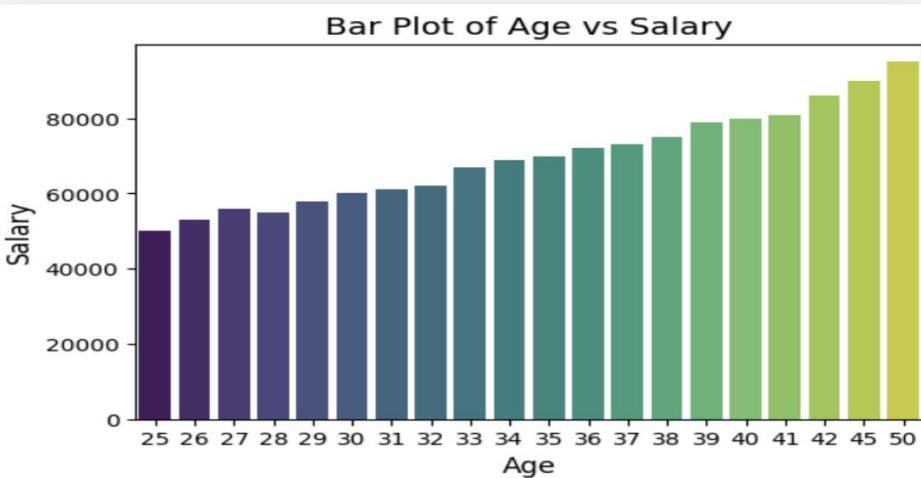
	Name	Age	Salary	Department	Experience (Years)
0	Alice	25	50000	IT	2
1	Bob	30	60000	HR	5
2	Charlie	35	70000	Finance	8
3	David	40	80000	IT	10
4	Eve	45	90000	Marketing	12
5	Frank	28	55000	HR	3
6	Grace	32	62000	Finance	6
7	Hannah	29	58000	IT	4
8	Ian	34	69000	HR	7
9	Jack	38	75000	Marketing	9

Dashboard page

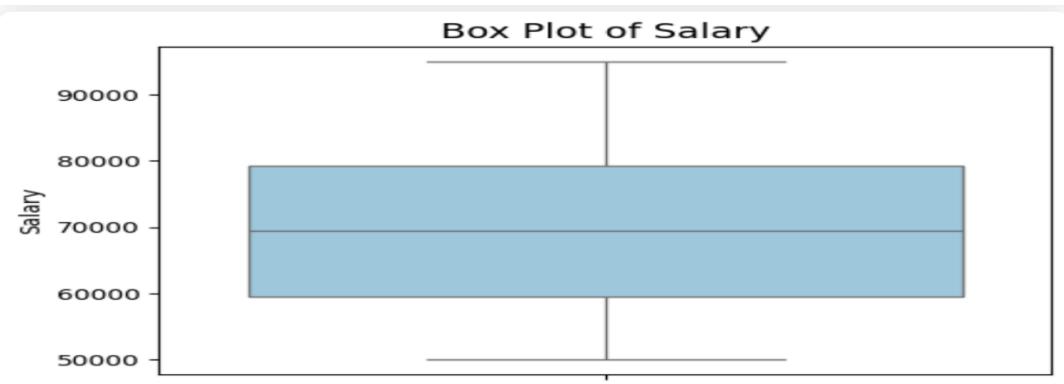
Data Visualizations



Histogram plot



Bar plot



[Upload Another File](#)

Box plot