Industrial Internship Report on

”Python”

Prepared by

Vishal Jadhav

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.

My project was

1. URL SHORTENER
2. FILE ORGANIZER
3. PASSWORD MANAGER DEVELOPMENT
4. QUIZ GAME

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

This report provides an overview of the 6-week Industrial Internship provided by Upskill Campus (USC) and The IoT Academy in collaboration with the Industrial Partner UniConverge Technologies Pvt Ltd (UCT). The internship aimed to enhance participants' Python programming skills and provide practical experience through various projects and learning materials.

About the need for a relevant internship in career development:

A relevant internship plays a crucial role in career development as it bridges the gap between theoretical knowledge and practical application. This internship focused on Python programming, allowing interns to apply their skills in real-world projects and gain valuable experience, which is essential for their professional growth.

Brief about your project/problem statement:

During the internship, I worked on multiple projects that spanned different areas of software development. One of the projects I focused on was a URL shortener. The objective of this project was to create a Python-based application that could convert long URLs into shorter, more manageable links. It involved designing a user interface, implementing a database for URL mapping, and developing functions for generating unique shortened URLs and handling redirection.

Another project I undertook was a file organizer. This Python project aimed to assist users in organizing their files within a directory. The program scanned a specified directory, categorized files based on their type, and moved them into respective folders. It required designing a user interface, implementing file type identification, and developing a file-moving algorithm.

Additionally, I worked on a password manager project. The goal was to develop a Python application that securely stored and managed user passwords. It involved implementing encryption algorithms, designing a user interface, and developing functions for generating strong passwords and storing/retrieving them from a database.

Lastly, I worked on a quiz game project that quizzed users on various topics. The Python program presented questions, collected user answers, and calculated scores. The project encompassed designing a user interface, implementing a database or file system for storing quiz data, and developing a scoring algorithm.

Opportunity given by USC/UCT:

Upskill Campus (USC) and The IoT Academy, in collaboration with UniConverge Technologies Pvt Ltd (UCT), provided a valuable opportunity to participate in this Industrial Internship. The internship offered

hands-on experience, exposure to real-world projects, and guidance from industry professionals, contributing to the interns' growth and development.

How the program was planned:

The program was well-planned to ensure a comprehensive learning experience. It incorporated both project-based work and self-study resources. Each week of the internship focused on specific topics, gradually building the participants' Python skills and knowledge. The program planning included providing the necessary project descriptions, guidance, and support materials to facilitate successful project completion.

Your learnings and overall experience:

Throughout the internship, I gained practical experience in Python programming and developed essential skills required in the industry. By working on the URL shortener project, I learned about web development, database management, and URL manipulation. The file organizer project enhanced my understanding of file handling and organization techniques. Developing the password manager project allowed me to delve into encryption algorithms and secure storage practices. Lastly, the quiz game project improved my skills in user interface design, data management, and score tracking. Overall, the internship provided a valuable learning experience that expanded my Python knowledge and prepared me for future endeavors in the field.

Thank you to all who have helped you directly or indirectly:

I would like to express my sincere gratitude to all the individuals who have directly or indirectly supported me throughout this internship:

- The organizers and coordinators at Upskill Campus and The IoT Academy for providing this opportunity.
- My fellow interns for their collaboration, shared learning experiences, and support.

Your message to your juniors and peers:

To my juniors and peers, I would like to share the following message:

"Embrace internship opportunities like this to enhance your skills, gain practical experience, and explore real-world projects. Be proactive in seeking knowledge, asking questions, and collaborating with your peers. Take advantage of the resources provided and engage in self-study to expand your understanding. Remember to stay curious, persistent, and open-minded. This internship has been a valuable stepping stone in my career journey, and I encourage you to actively pursue similar opportunities to grow and succeed."

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

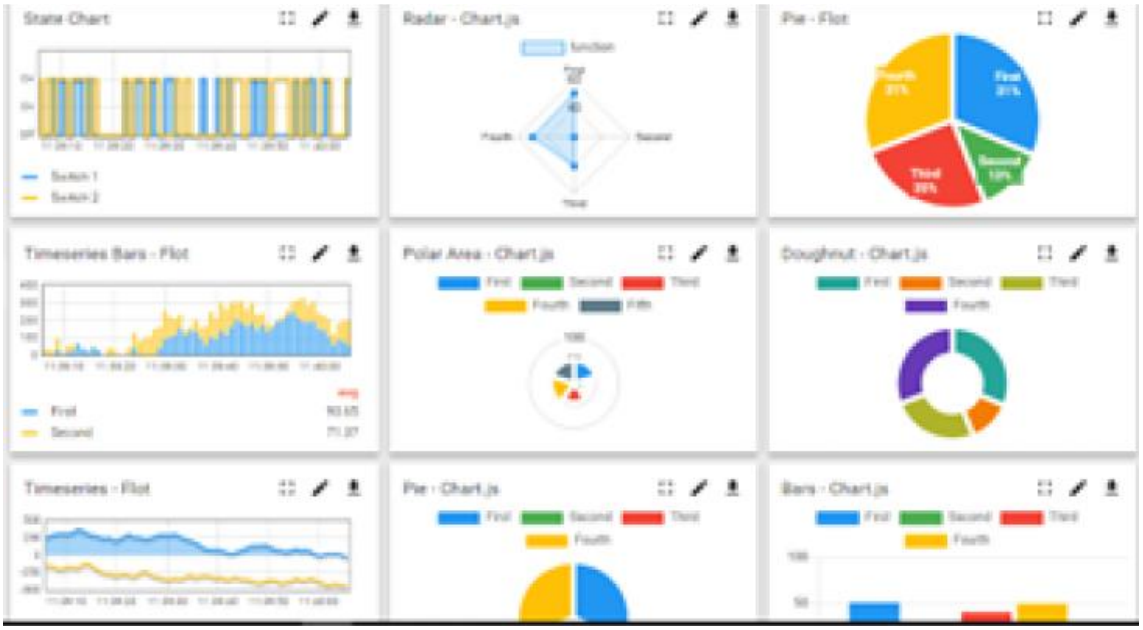
i. UCT IoT Platform (UCT Insight)

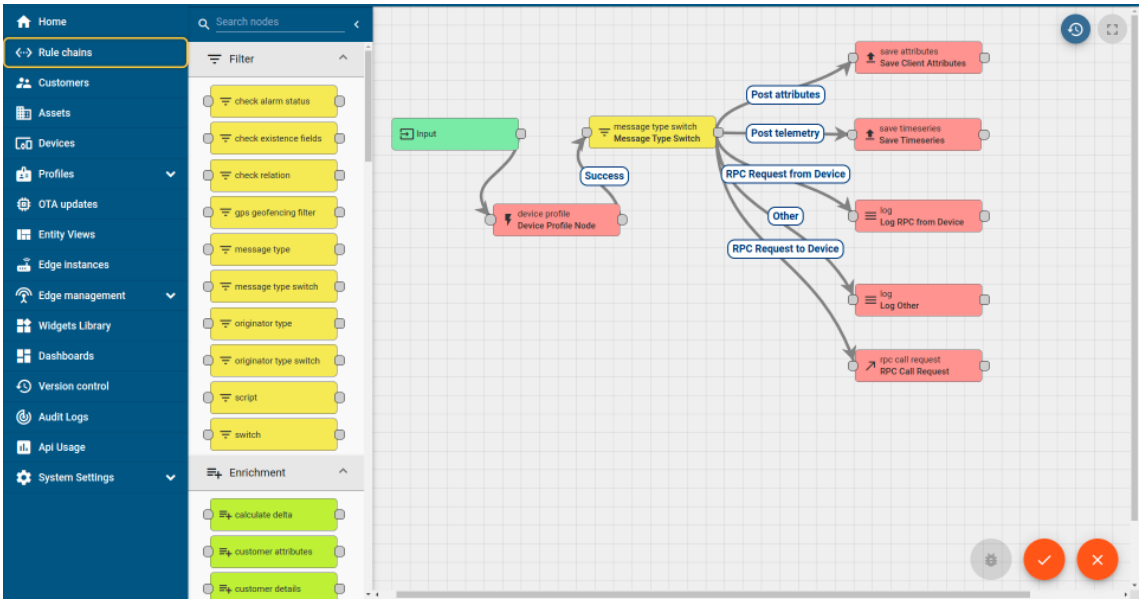
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

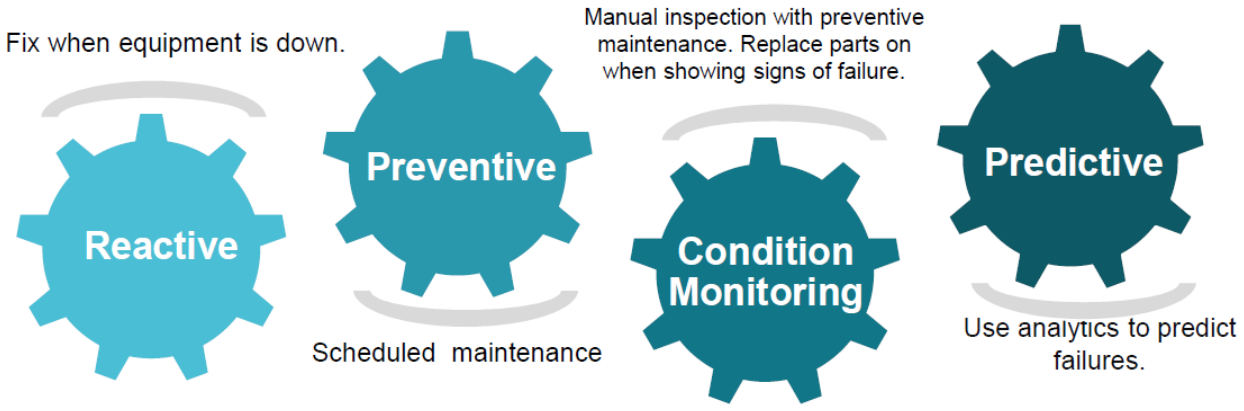
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

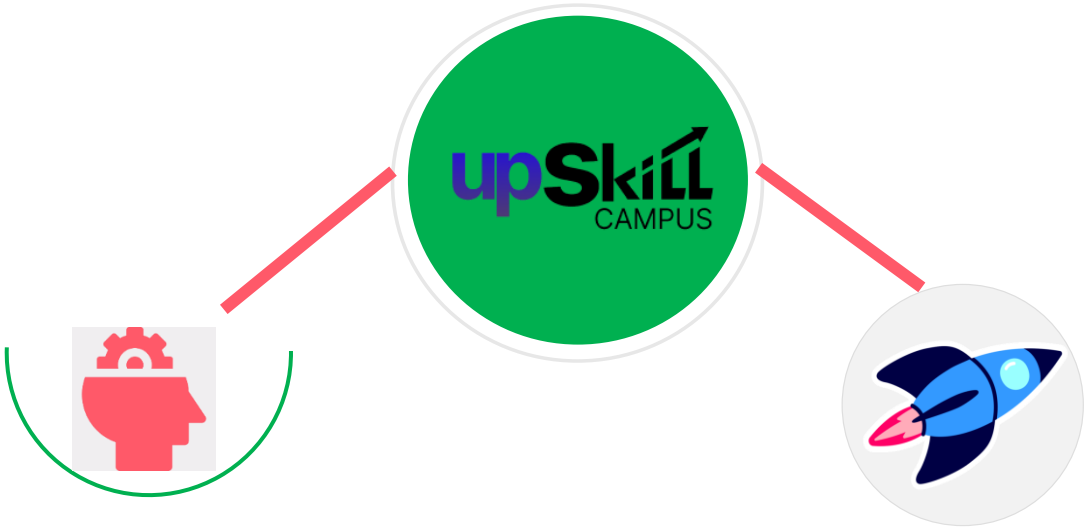
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

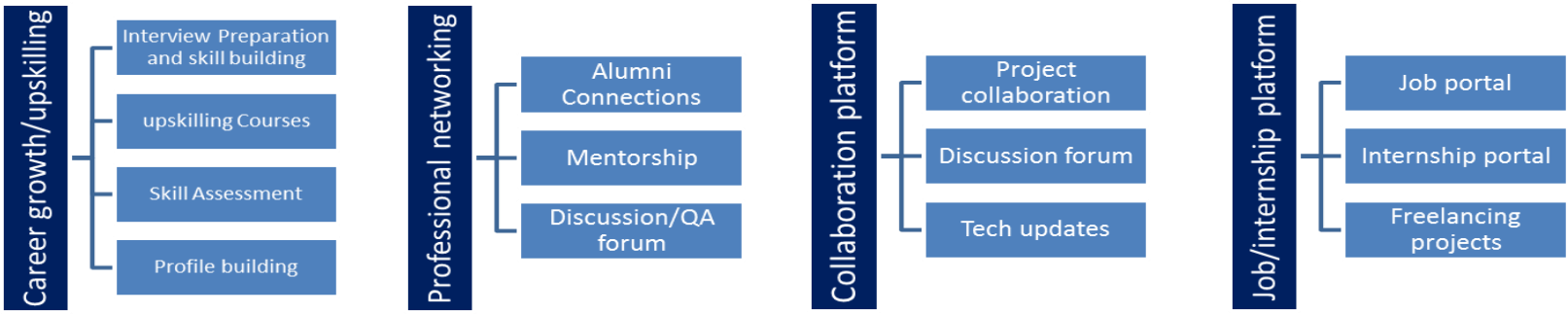
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] "Python for Everybody" by Charles R. Severance

[2] " Learning Python: Powerful Object-Oriented Programming " by Mark Lutz

[3] NumPy documentation: https://numpy.org/doc/

[4] Pandas documentation: https://pandas.pydata.org/docs/

2.6 Glossary

Terms	Acronym
GUI	Graphical User Interface
URL	Uniform Resource Locator
SQLite	Structured Query Language Lite
Fernet	Symmetric Encryption Algorithm
API	Application Programming Interface
Flask	Python Web Framework
tkinter	Python GUI Library
NumPy	Numerical Python

3 Problem Statement

Problem Statement 1: URL Shortener

Summary: The problem is to develop a Python project that converts long URLs into shorter, more manageable links. It should take a long URL as input, generate a unique shortened URL, and redirect users to the original URL when the shortened link is accessed.

Problem Statement 2: File Organizer

Summary: The problem is to develop a Python project that helps users organize their files in a directory. It should scan a specified directory, categorize files based on their type, and move them into respective folders.

Problem Statement 3: Password Manager

Summary: The problem is to develop a Python project that securely stores and manages user passwords. It should allow users to store passwords for various accounts, generate strong passwords, and retrieve passwords when needed.

Problem Statement 4: Quiz Game

Summary: The problem is to develop a Python project that quizzes users on various topics. It should read questions and answers from a file or database, present them to the user, and keep track of their score.

4 Existing and Proposed solution

1 URL Shortener

Existing Solutions and Limitations:

There are existing URL shortener services available, such as Bitly and TinyURL. These services offer the functionality of converting long URLs into shorter ones. However, they have limitations such as dependency on external platforms, potential privacy concerns, and limited customization options.

Proposed Solution:

The proposed solution is to develop a Python-based URL shortener project that provides a self-hosted and customizable solution. By creating a standalone application, we eliminate the reliance on external platforms and provide more control over data privacy. The solution will generate unique shortened URLs and ensure seamless redirection to the original URLs.

Value Addition:

The proposed solution aims to address the limitations of existing URL shortener services. By offering a self-hosted solution, users have more control over their data and privacy. Additionally, the project will provide customization options, allowing users to customize the shortened URLs according to their preferences. This increased flexibility and control are the key value additions of the proposed solution.

2 File Organizer

Existing Solutions and Limitations:

There are file organizing tools and utilities available that automate file organization. These solutions often rely on predefined rules and patterns to categorize files. However, they may have limitations in terms of flexibility and customization options. Some existing solutions also lack user-friendly interfaces and may not support all file types.

Proposed Solution:

The proposed solution is to create a Python-based file organizer project that offers flexibility and customization options. The solution will scan a specified directory, identify file types using their extensions, and move them into respective folders based on user-defined rules. It will provide a user-friendly interface to specify the directory, set rules, and perform the organization process efficiently.

Value Addition:

The proposed solution enhances the flexibility and customization options for file organization. Users can define their own rules and categories to suit their specific needs. Additionally, the user-friendly interface simplifies the organizing process and makes it accessible to users with varying technical expertise. These value additions make the proposed solution stand out from existing file organizing tools.

3 Password Manager

Existing Solutions and Limitations:

There are password manager applications available that offer password storage and management. These applications often provide encryption algorithms to secure passwords. However, some existing solutions may have limitations in terms of platform compatibility, usability, or limited customization options.

Proposed Solution:

The proposed solution is to create a Python-based password manager project that provides secure password storage and management. It will implement encryption algorithms to ensure the privacy of stored passwords. The solution will offer a user-friendly interface to store, retrieve, and generate strong passwords for different accounts.

Value Addition:

The proposed solution aims to provide a user-friendly and secure password management system. By offering a Python-based solution, it ensures platform compatibility and flexibility. The project will prioritize data privacy and encryption to enhance the security of stored passwords. Additionally, the user-friendly interface will make it easy for users to manage their passwords effectively.

4. Quiz Game

Existing Solutions and Limitations:

There are quiz game applications available that provide question-and-answer functionalities. These solutions often offer predefined question sets and limited customization options. Some existing solutions may also lack flexibility in terms of adding new questions or supporting multiple question formats.

Proposed Solution:

The proposed solution is to create a Python-based quiz game project that offers flexibility and customization options. The solution will read questions and answers from a file or database, allowing easy addition and modification of question sets. It will provide a user-friendly interface to present questions, collect user answers, and calculate the user's score.

Value Addition:

The proposed solution provides flexibility and customization options for creating and managing quiz games. By reading questions from a file or database, it allows easy updates and additions to the question set. The user-friendly interface enhances the user experience and provides a seamless quiz-taking process. These value additions make the proposed solution more versatile compared to existing quiz game applications.

4.1 Code submission

https://github.com/vishal117jadhav/UpSkillCampus

4.2 Report submission

https://github.com/vishal117jadhav/UpSkillCampus

5 Proposed Design/ Model

The design flow for the URL Shortener solution using Flask:

1. Start:

- The solution starts by importing the necessary libraries: Flask, string, random, and sqlite3.
- A thread-local storage is created to handle the database connection and cursor for each thread.
- The Flask application is initialized.

2. Intermediate Stages:

- The `get_database_connection` function is defined to create and return the database connection for the current thread.
- The `get_database_cursor` function is defined to create and return the database cursor for the current thread.
- The `close_database_connection` function is defined to close the database connection and cursor after the request is completed.
- The `before_request` decorator is used to create the database connection and cursor before each request.
- The `teardown_request` decorator is used to close the database connection and cursor after each request.
- The `create_url_mapping_table` function is defined to create the URL mapping table in the database if it doesn't already exist.
- The `generate_shortened_url` function is defined to generate a random short URL using alphanumeric characters.
- The `'/'` route handles both GET and POST requests. For a GET request, it retrieves the URL mappings from the database and renders the template with the mappings. For a POST request, it receives the long URL, generates a short URL, inserts the mapping into the database, and redirects back to the homepage.
- The `'/<short_url>'` route handles requests with a short URL as a parameter. It retrieves the long URL from the database based on the short URL and redirects the user to the original URL.

3. Final Outcome:

- The Flask application is run with the `create_url_mapping_table` function called to create the URL mapping table if it doesn't exist.
- The application handles requests, manages the database connection and cursor for each thread, and performs the URL shortening and redirection functionalities.

The design flow for the File Organizer solution:

1. Start:

- The solution starts by importing the required libraries: os, shutil, and tkinter.
- The necessary functions and variables are defined.

2. Intermediate Stages:

- The `organize_files` function is defined to handle the organization of files within a directory.
- The `file_types` dictionary is created, mapping file categories to their corresponding file extensions.
- The function prompts the user to select a directory where the files will be organized.
- The function checks if the necessary folders for file categories exist. If not, it creates them.
- The function scans the selected directory and moves files to the appropriate folders based on their extensions.

3. GUI Creation:

- The `select_directory` function is defined as the command for the "Select Directory" button.
- When the button is clicked, the function opens a file dialog for the user to choose the directory.
- If a directory is selected, the `organize_files` function is called with the chosen directory as the input.
- After organizing the files, a success message is printed.

4. Main Execution:

- The main window for the GUI is created using `tkinter`.
- A button widget is created for selecting the directory, and it is associated with the `select_directory` function.
- The GUI main loop is run to display the window and handle user interactions.

The design flow ensures that the necessary libraries are imported, the functions are defined in the appropriate order, and the GUI elements are created. The intermediate stages involve the core logic of organizing files, including handling file extensions, creating folders, and moving files. The final outcome is the functioning GUI application that allows users to select a directory and organize their files with the click of a button.

The design flow for the Password Manager solution:

1. Start:

- The solution starts by importing the necessary libraries: sqlite3, bcrypt, cryptography, getpass, and random.
- The database connection is established with the passwords.db file.
- The table for storing passwords is created if it doesn't already exist.
- The Fernet key for encryption and decryption is generated.

2. Intermediate Stages:

- The solution provides various functions to perform password management tasks:
- `generate_password`: Generates a random password using alphanumeric characters and punctuation.
- `encrypt_password`: Encrypts a password using the Fernet key.
- `decrypt_password`: Decrypts an encrypted password using the Fernet key.
- `save_password`: Inserts an account, username, and encrypted password into the database.
- `retrieve_password`: Retrieves a username and decrypted password from the database based on the account.
- `generate_and_save_password`: Generates a random password, saves it with the account and username in the database, and returns the generated password.
- `main_menu`: Displays the main menu options, prompts for user input, and calls the respective functions based on the choice.

3. Final Outcome:

- The main menu is displayed, allowing the user to perform password management tasks:
- Option 1: Add a password - The user enters the account name, username, and password to be saved.
- Option 2: Retrieve a password - The user enters the account name and retrieves the corresponding username and decrypted password from the database.
- Option 3: Generate and save a password - The user enters the account name and username, generates a random password, saves it in the database, and displays the generated password.
- Option 4: Exit - The program terminates.

The design flow ensures that the necessary libraries are imported, the database connection is established, and the required functions are defined in the appropriate order. The intermediate stages involve the core logic of password encryption, decryption, saving, retrieval, and generation. The final outcome is a functioning password manager program that allows users to add, retrieve, and generate passwords with a user-friendly menu.

The design flow for the Quiz Game solution:

1. Start:

- The solution starts by defining the `QuizGame` class and importing the necessary libraries, such as `random` for shuffling the questions.

2. Intermediate Stages:

- The `QuizGame` class is defined with an `__init__` method that initializes the game with a list of questions and a starting score of 0.
- The `start_game` method is defined to begin the quiz game. It shuffles the questions, iterates over each question, displays the question text and options (if available), prompts the user for an answer, checks the answer for correctness, and updates the score.
- The `display_options` method is defined to display the available options for a multiple-choice question.
- The `check_answer` method is defined to compare the user's answer with the correct answer and update the score accordingly.
- The `display_final_score` method is defined to display the final score after all questions have been answered.

3. Final Outcome:

- The quiz questions are defined as a list of dictionaries, each representing a question with its text, options, and correct answer.
- An instance of the `QuizGame` class is created with the list of questions.
- The `start_game` method is called to begin the quiz game.

The design flow ensures that the necessary classes, methods, and data structures are defined in the correct order. It allows for the initialization of the game, handling of question display and user input, checking of answers, and displaying the final score.

6 Performance Test

6.1 Test Plan/ Test Cases

URL Shortener:

- Constraint: Memory Usage
- Test Case 1: Input a large number of URLs and measure the memory consumption during the URL shortening process. Ensure that memory usage remains within acceptable limits.
- Test Case 2: Simulate concurrent requests to the URL shortener and measure memory usage to ensure scalability and efficient memory management.

File Organizer:

- Constraint: Speed (MIPS - Million Instructions Per Second)
- Test Case 1: Measure the time taken to organize a directory containing a large number of files. Ensure that the file organization process is efficient and completed within an acceptable timeframe.
- Test Case 2: Test the file organizer with different file types and sizes to identify any performance bottlenecks. Optimize the algorithm to handle files of various types and sizes efficiently.

Password Manager:

- Constraint: Data Security (Durability)
- Test Case 1: Store a large number of passwords and verify the durability of the password storage mechanism. Ensure that the encrypted passwords are securely stored and can be retrieved without any loss of data.
- Test Case 2: Test the password manager with different encryption algorithms and key management strategies to evaluate their impact on data security and durability.

Quiz Game:

- Constraint: Accuracy
- Test Case 1: Verify that the quiz game accurately tracks and calculates the user's score based on their answers to the questions.
- Test Case 2: Test the quiz game with a variety of question types and answer formats to ensure accurate handling of different question scenarios.

6.2 Test Procedure

- For each solution (URL Shortener, File Organizer, Password Manager, Quiz Game), create a test plan detailing the identified constraints, test cases, and expected outcomes.
- Implement automated tests using appropriate testing frameworks or scripts to execute the test cases and capture the results.
- Execute the test procedures for each solution and record the test results, including any issues or observations encountered during testing.

6.3 Performance Outcome

- Measure and analyze the performance outcomes of each solution based on the identified constraints (e.g., memory usage, speed, data security, accuracy).
- Document the performance results, including any performance improvements or optimizations implemented to meet the constraints.
- Provide recommendations for handling any identified constraints that were not fully tested, including potential impacts on the design and suggestions for mitigating those constraints.

7 My learnings

Summary of Overall Learning and Career Growth:

Throughout the six weeks of this industrial internship, I have gained valuable knowledge and practical experience in various aspects of software development. The internship provided me with an opportunity to work on real-world projects, collaborate with industry professionals, and apply my technical skills in a professional setting. This experience has greatly contributed to my career growth in the following ways:

1. Technical Skills Enhancement:

- I have acquired proficiency in Python programming language and gained hands-on experience in developing applications using different libraries and frameworks.
- I have expanded my knowledge of web development by working with Flask framework and building a URL shortener application.
- I have learned database management techniques and utilized SQLite to store and retrieve data in the password manager project.
- I have gained experience in working with file systems, organizing files, and developing a user-friendly interface in the file organizer project.
- I have developed a quiz game application, enhancing my skills in handling data structures, user interactions, and scoring algorithms.

2. Problem-Solving and Critical Thinking:

- Working on diverse projects challenged me to think critically and develop innovative solutions to complex problems.
- I learned to analyze requirements, break down tasks, and implement effective solutions using logical and systematic approaches.
- Debugging and troubleshooting various issues throughout the internship improved my problem-solving skills and attention to detail.

3. Collaboration and Communication:

- I learned to communicate ideas, discuss project requirements, and seek feedback to improve the quality of my work.

4. Career Growth:

- The practical experience gained during this internship has given me a solid foundation in software development and increased my confidence in tackling real-world projects.
- The exposure to industry-standard tools, technologies, and practices has prepared me to meet the demands of the professional world.
- The internship has provided me with a strong portfolio and practical examples to showcase to potential employers, enhancing my job prospects in the field of software development.

Overall, this industrial internship has been a transformative experience, enabling me to acquire new skills, enhance my knowledge, and grow both personally and professionally. I am confident that the learning and experiences gained during this internship will significantly contribute to my career growth and future success in the software development industry.

8 Future work scope

While the industrial internship provided valuable learning opportunities and practical experience, there were some ideas and areas that could not be explored fully due to time limitations. Here are some potential future work scope ideas that could be pursued:

1. Advanced Features for the URL Shortener:
   - Implement custom URL aliases to allow users to choose their own shortened URLs.
   - Enhance the redirection mechanism by implementing analytics and tracking features.
   - Integrate URL expiration functionality to automatically expire shortened links after a certain period.

2. File Organizer Enhancements:
   - Develop additional file categorization algorithms based on file content analysis, rather than relying solely on file extensions.
   - Implement support for custom file organization rules and configurations to cater to individual user preferences.
   - Extend the file organizer to handle file renaming, metadata extraction, and advanced search capabilities.

3. Password Manager Improvements:
   - Integrate password strength analysis and provide recommendations for generating stronger passwords.
   - Implement multi-factor authentication for enhanced security.
   - Add support for securely syncing passwords across multiple devices and platforms.

4. Quiz Game Enhancements:
   - Implement a timer feature to introduce time constraints for answering questions.
   - Introduce difficulty levels and categories to provide a personalized and challenging quiz experience.
   - Develop a leaderboard or scoring system to track and compare users' scores.

5. Performance Optimization:
   - Identify performance bottlenecks in each project and optimize algorithms or data structures to improve efficiency.
   - Implement caching mechanisms to reduce database or file system access and improve response times.
   - Conduct scalability testing to ensure the solutions can handle increased user load and larger datasets.

These future work scope ideas provide avenues for further exploration and improvement in the respective projects. They offer opportunities to enhance functionality, security, user experience, and performance. Pursuing these ideas in the future could lead to more robust and feature-rich applications in line with industry requirements and user expectations.