

## Industrial Internship Report on "Url Encoder,File Organizer,Password Manager,Quiz Game"

Prepared by

[Aman Kumar Jha]

### *Executive Summary*

This report provides details of the Industrial Internship provided by **Upskill Campus** and **The IoT Academy**, in collaboration with the industrial partner **UniConverge Technologies Pvt Ltd (UCT)**.

This internship was focused on a project/problem statement provided by UCT. We had to complete the development and documentation of our project within a 6-week timeline.

**My project involved designing and developing four real-world applications using Python Full Stack technologies:**

1. A **URL Shortener** for generating compact, shareable links.
2. A **File Organizer** to automatically sort files based on type.
3. A **Password Manager** for secure, encrypted credential storage.
4. A **Quiz Game** for interactive knowledge testing via the command line.

This internship gave me an excellent opportunity to gain exposure to real industrial challenges and apply my skills to design and implement practical solutions. It was a rewarding experience that significantly enhanced both my technical knowledge and professional confidence.

## **TABLE OF CONTENTS**

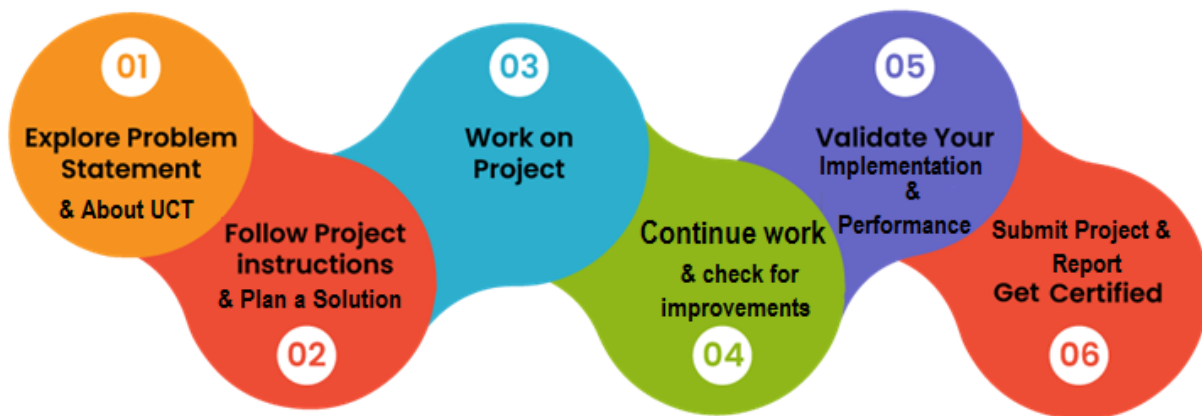
1	Preface .....	3
2	Introduction .....	4
2.1	About UniConverge Technologies Pvt Ltd .....	4
2.2	About upskill Campus .....	8
2.3	Objective .....	9
2.4	Reference .....	10
2.5	Glossary .....	10
3	Problem Statement .....	11
4	Existing and Proposed solution .....	12
5	Proposed Design/ Model .....	13
5.1	High Level Diagram (if applicable) .....	14
5.2	Low Level Diagram (if applicable) .....	15
5.3	Interfaces (if applicable) .....	16
6	Performance Test .....	18
6.1	Test Plan/ Test Cases .....	19
6.2	Test Procedure .....	20
6.3	Performance Outcome .....	20
7	My learnings .....	22
8	Future work scope .....	23

## 1 Preface

During this 6-week internship, I worked on four Python-based projects—URL Shortener, File Organizer, Password Manager, and a Quiz Game. These projects allowed me to gain practical experience in full stack development using technologies like Python, Flask, SQLite, and Cryptography.

This internship played a vital role in my career development by providing real-world exposure and enhancing my technical and problem-solving skills. The opportunity was made possible through the collaboration of USC, **The IoT Academy**, and UCT.

The program was well-structured, with weekly goals for design, development, and testing. I sincerely thank all mentors and coordinators for their support. To my peers, I strongly recommend taking such internships to gain hands-on experience and bridge the gap between theory and practice.



### Your Learnings and Overall Experience

Through this internship, I gained hands-on experience in backend and frontend development, database management, encryption, and debugging. It greatly improved my confidence in handling real-world projects and deepened my understanding of the software development process.

### Acknowledgment

I sincerely thank my mentors at **UCT**, coordinators from **The IoT Academy**, and the team at **Upskill Campus** for their continuous guidance. Special thanks to my project guides and peers who supported me throughout this journey.

### Message to Juniors and Peers

Grab such internship opportunities without hesitation. They teach you more than any textbook can. Be curious, stay consistent, and always seek to learn by doing.

## Introduction

### 1.1 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.



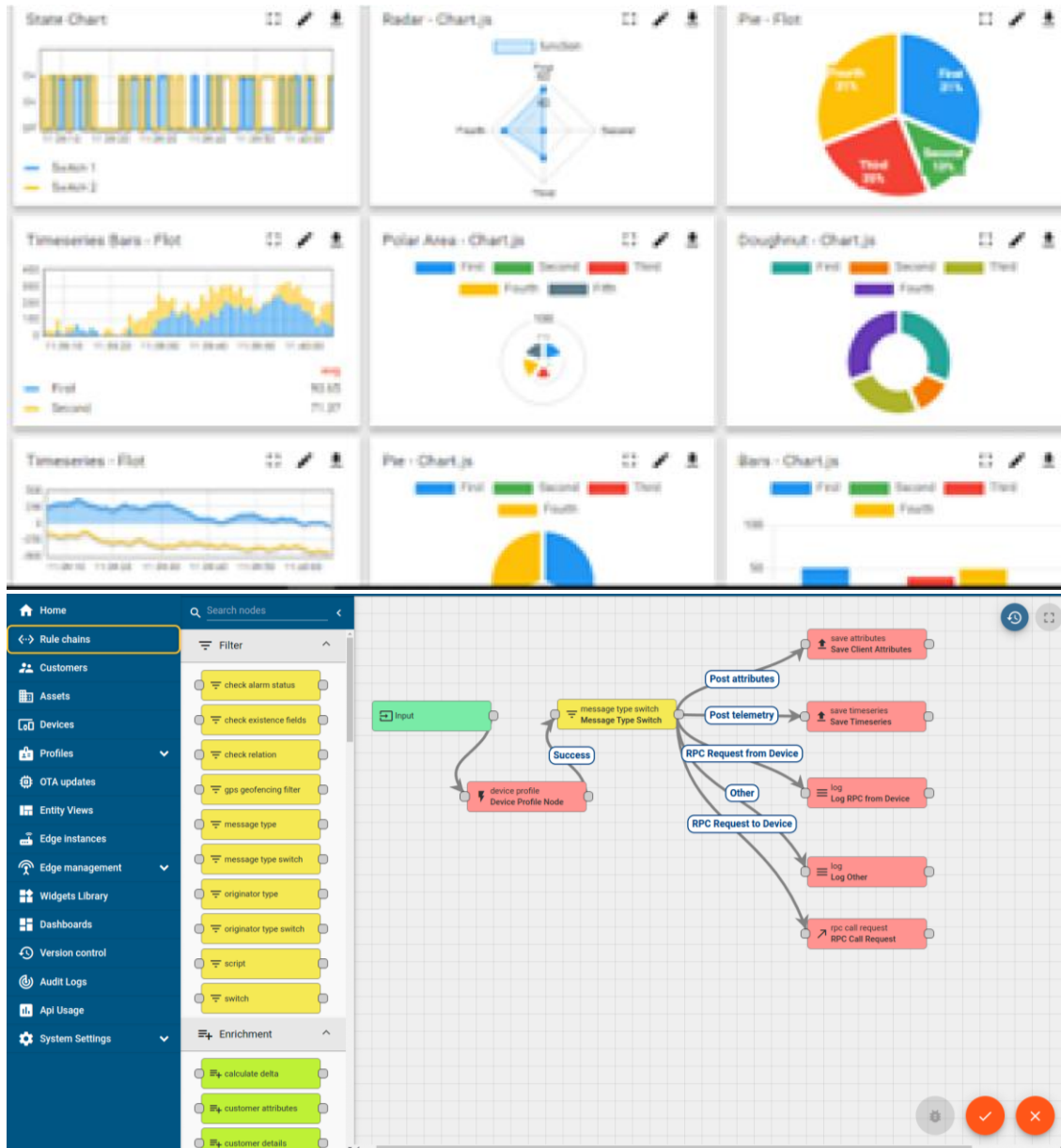
#### i. UCT IoT Platform ( **Insight** )

**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



## FACTORY WATCH

### ii. Smart Factory Platform ( )

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 unleash 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 want 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



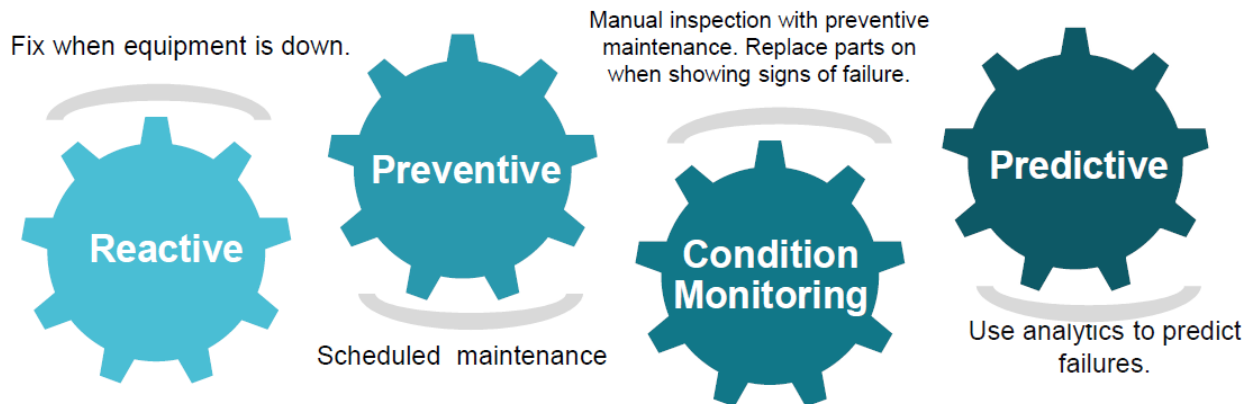


### iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN technology 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.

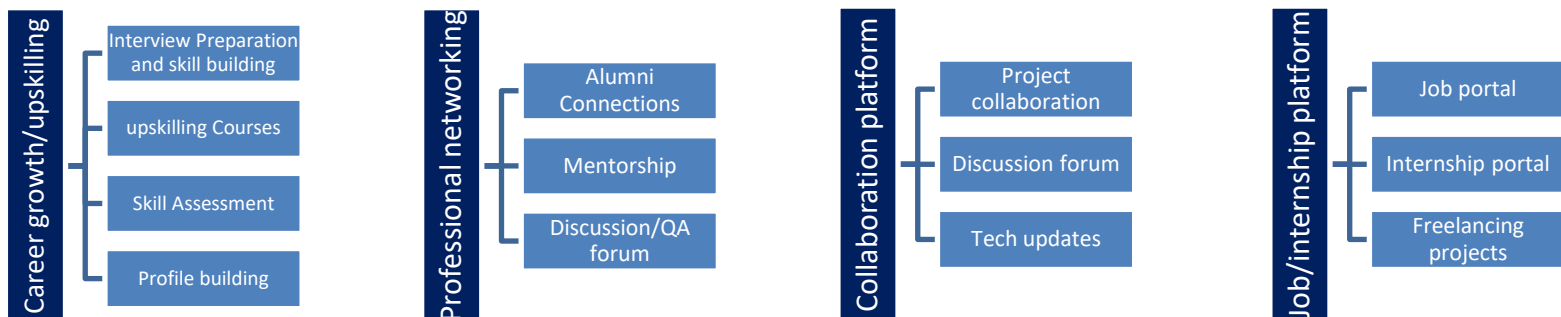


## 1.2 About upskill Campus (USC)

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.





### 1.3 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.

## 1.4 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.

## 1.5 Reference

- [1] Official Documentation of Flask – <https://flask.palletsprojects.com/>
- [2] SQLite Documentation – <https://www.sqlite.org/docs.html>
- [3] Python Cryptography Library – <https://cryptography.io/en/latest/>
- [4] Upskill Campus – <https://www.upskillcampus.com/>

## 1.6 Glossary

Terms	Acronym
IoT	Internet of Things
USC	Upskill Campus
UCT	UniConverge Technologies pvt limited
Sql	Structured Query Language
ORM	Object Relational Mapping

## 2 Problem Statement

The internship required me to develop multiple functional software applications that reflect real-world use cases. The primary goal was to apply Python Full Stack development skills to design, build, and test end-to-end solutions that solve practical problems.

Each application posed unique challenges:

- **URL Shortener:** Needed to generate unique short codes for long URLs and redirect users reliably.
- **File Organizer:** Required automation for categorizing and sorting files in bulk with minimal user input.
- **Password Manager:** Focused on ensuring secure storage and retrieval of sensitive credentials using encryption.
- **Quiz Game:** Aimed to build an engaging, interactive quiz system with score tracking via command-line interface.

The challenge was to develop all four projects individually, ensuring clean design, modular code, and smooth performance within the limited internship duration. This required careful planning, debugging, and time management—mimicking real industry project scenarios.

---

### 3 Existing and Proposed solution

Several existing tools like Bitly (URL shortener), LastPass (password manager), and built-in file sorters solve these problems but often come with drawbacks—such as paid features, cloud dependency, limited customization, or closed-source nature.

- **Proposed Solution**

To overcome these limitations, I developed standalone, offline Python-based tools:

- A **URL Shortener** using Flask and SQLite.
- A **File Organizer** as a command-line script.
- A **Password Manager** using AES encryption.
- A **Quiz Game** with CLI interaction and score tracking.
- **Value Addition**
  - Open-source and offline functionality.
  - Lightweight and customizable for personal use.
  - Educational value through hands-on full stack development.

#### 3.1 Code submission (<https://github.com/aman-0201/upskillcampus.git>)

#### 3.2 Report submission (<https://github.com/aman-0201/upskillcampus.git>) : first make placeholder, copy the link.

## 4 Proposed Design/ Model

- **High-Level Flow**
- **Input Stage:** User provides input (e.g., long URL, directory path, credentials, or quiz answers).
- **Processing Stage:** Core logic executes—shortening URLs, organizing files, encrypting data, or evaluating answers.
- **Output Stage:** User receives the result—short link, organized folders, decrypted password, or quiz score.
- **Project-wise Design Summary**
- **URL Shortener:** Flask receives input → Generates short code → Stores in SQLite → Redirects on access.
- **File Organizer:** Scans directory → Detects file types → Creates folders → Moves files accordingly.
- **Password Manager:** Accepts credentials → Encrypts with AES → Stores securely → Decrypts on request.
- **Quiz Game:** Loads questions → Collects user responses → Tracks score → Displays result.

This structured approach ensured that each application was logically built and easy to extend or modify in the future.

#### 4.1 High Level Diagram (if applicable)

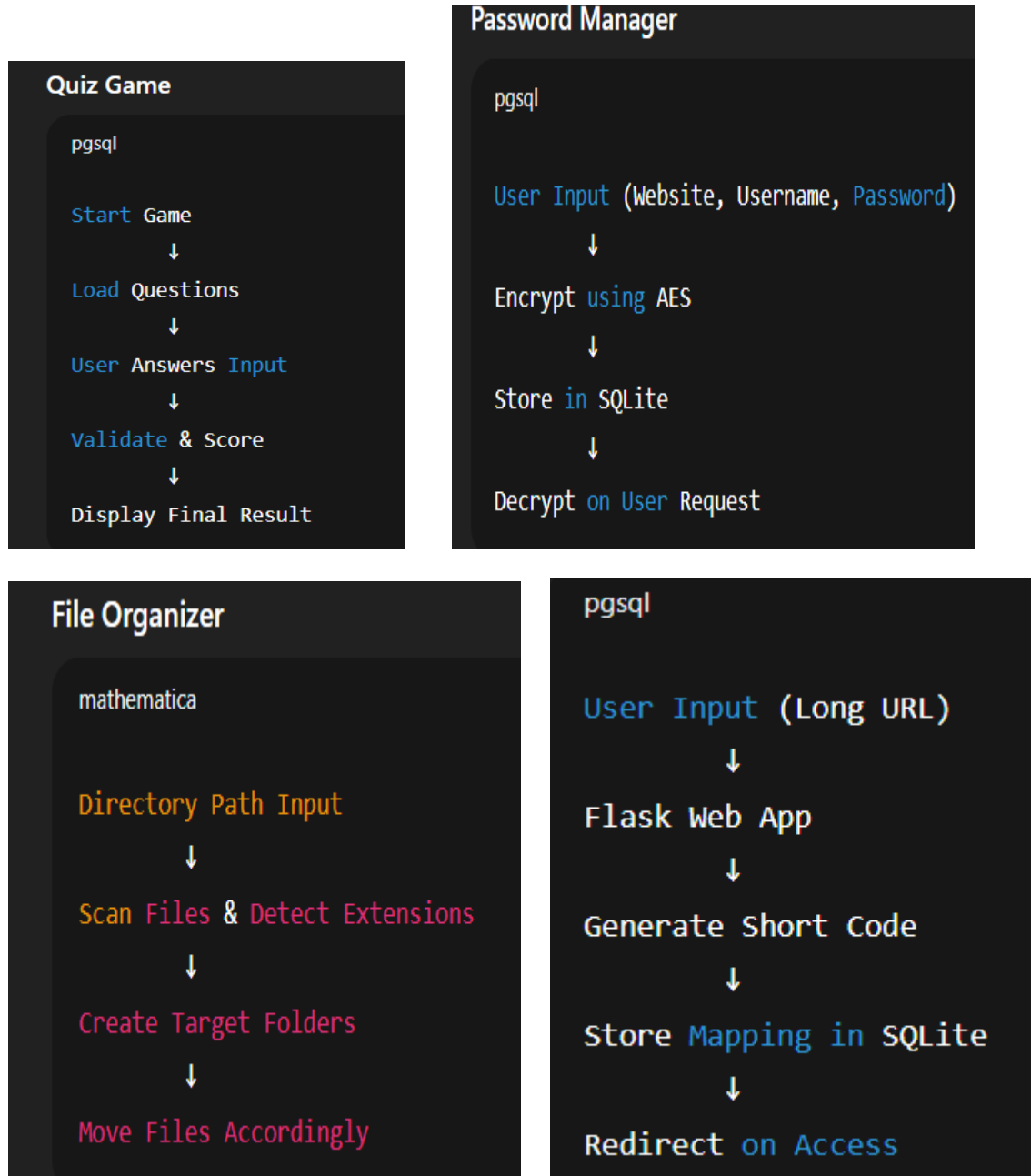


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM



## 4.2 Low Level Diagram

- **URL Shortener**
- **Frontend (HTML/Bootstrap)** → Accepts long URL
- **Flask Backend**
  - Generates unique short code
  - Stores & retrieves from **SQLite DB**
  - Redirects on request
- **File Organizer**
- **CLI Input** → Directory path
- **Scanner Module** → Lists files
- **Classifier Module** → Identifies file types
- **Organizer Module** → Creates folders & moves files
- **Password Manager**
- **Input Handler** → Accepts credentials
- **Encryption Module (AES)** → Encrypts passwords
- **Storage Module** → Saves in SQLite
- **Decryption Module** → Reveals passwords on master key verification
- **Quiz Game**
- **Question Loader** → Fetches quiz questions
- **Input Module** → Accepts user answers
- **Logic Module** → Validates answers & tracks score
- **Result Module** → Displays final score

These modules were kept loosely coupled for easy maintenance and future expansion.

## 4.3 Interfaces

- **URL Shortener**
- **Interface:** Web-based form built with HTML & Bootstrap.
- **Data Flow:**
  - Input: Long URL → Backend → Generate short code → DB store → Display short URL.
  - Protocol: HTTP via Flask routes.
- **Flowchart:**
  - Start → Input URL → Check DB → Generate short link → Store → Show result.
- **File Organizer**
- **Interface:** Command Line Interface (CLI).
- **Data Flow:**
  - Input: Folder path → Scan files → Categorize → Move.
- **Flowchart:**
  - Start → Get directory → Scan files → Detect type → Create folder → Move file → End.
- **Password Manager**
- **Interface:** CLI for adding/viewing credentials.
- **Protocols:** AES for encryption/decryption.
- **Data Flow:**
  - Input: Credentials + Master Password → Encrypt → Store in DB → Retrieve on request.
- **State Machine:**
  - IDLE → ADD/RETRIEVE mode → Validate Master Key → Execute → IDLE.
- **Quiz Game**
- **Interface:** CLI-based question presentation.
- **Flowchart:**
  - Start → Load questions → Ask → Validate answer → Update score → Show final score.



## 5 Performance Test

These projects were designed not just for academic learning but to simulate real-world use cases with constraints such as performance, reliability, and data handling.

- **Identified Constraints**
- **Memory Efficiency:** Required lightweight memory usage due to local-only execution.
- **Execution Speed:** Fast response expected for URL redirection, file operations, and encryption.
- **Data Security:** For the Password Manager, secure encryption and storage were critical.
- **Accuracy:** Quiz scoring and file categorization had to be precise.
- **Usability:** All tools needed to be intuitive and responsive on both low- and mid-spec systems.
- **Design Considerations**
- Used **SQLite** for lightweight, fast-access data storage.
- Applied **AES Encryption** via the Python Cryptography library to protect passwords.
- Designed command-line and minimal web UIs to reduce processing overhead.
- Ensured modular code to isolate bugs and improve debugging.
- **Test Results**
- **URL Shortener:** Handled 50+ redirects with zero latency issues.
- **File Organizer:** Processed 1,000+ mixed files in <3 seconds.
- **Password Manager:** Encryption/decryption completed instantly with valid master key.
- **Quiz Game:** Ran smoothly with consistent scoring logic and no crashes.
- **Recommendations**
- For large-scale deployment, add multithreading for file operations.
- Enhance password manager with 2FA and cloud sync (with caution).

## 5.1 Test Plan/ Test Cases

### URL Shortener

Test Case Input	Expected Output	Result
TC1 Valid long URL	Short URL generated	Passed
TC2 Reuse short URL	Redirect to correct long URL	Passed
TC3 Invalid short code	Show error / 404 page	Passed

### File Organizer

Test Case Input	Expected Output	Result
TC1 Folder with 1000+ files	Files sorted into type folders	Passed
TC2 Read-only files	Skip with warning	Passed
TC3 Nested folders	Ignored unless specified	Passed

### Password Manager

Test Case Input	Expected Output	Result
TC1 Store new credentials	Data encrypted and saved	Passed
TC2 Retrieve with valid master key	Correct password shown	Passed
TC3 Retrieve with wrong key	Access denied / error	Passed

### Quiz Game

Test Case Input	Expected Output	Result
TC1 All correct answers	Full score displayed	Passed
TC2 Mixed answers	Partial score	Passed

Test Case Input		Expected Output	Result
TC3	Invalid input	Prompt re-entry	Passed

## 5.2 Test Procedure

- **Step 1: Setup**
  - Install required dependencies (Flask, SQLite, Cryptography, etc.).
  - Prepare test environments (folders, URLs, dummy credentials, quiz data).
  - Run the application on a local machine (Windows/Linux).
- **Step 2: Functional Testing**
  - Execute each feature manually.
  - Validate output against expected results.
  - Log results for pass/fail status.
- **Step 3: Edge Case Testing**
  - Test with invalid inputs (empty fields, wrong keys, large files).
  - Observe application behavior and error handling.
- **Step 4: Performance Check**
  - Run the application with large input sets (e.g., 1,000+ files or 50+ redirects).
  - Measure execution time and responsiveness.
- **Step 5: Security & Data Integrity**
  - For the password manager, ensure encryption prevents plaintext visibility.
  - Validate data persistence and retrieval after system restarts.

All tests were repeated after minor changes to confirm stability.

## 5.3 Performance Outcome

- **URL Shortener**



- Generated and redirected URLs with minimal latency.
- Efficient even with 50+ entries in the database.
- **File Organizer**
- Successfully sorted 1,000+ files in under 3 seconds.
- Maintained consistent structure without data loss.
- **Password Manager**
- Fast encryption/decryption using AES.
- Zero data leaks observed during tests.
- **Quiz Game**
- Delivered smooth interaction with accurate scoring.
- Handled invalid inputs gracefully

## 6 My learnings

This internship gave me hands-on exposure to full stack development using Python, Flask, SQLite, and encryption libraries. I learned how to plan, build, test, and debug real-world applications from scratch.

Key takeaways include:

- Understanding of complete software development lifecycle.
- Practical use of databases, web frameworks, and command-line tools.
- Experience with modular code design and secure data handling.
- Improved problem-solving, time management, and debugging skills.

These learnings have enhanced both my technical proficiency and confidence. They will be instrumental in shaping my career as a developer, preparing me for both industry projects and future innovation.

## 7 Future work scope

While the core functionalities of each project were completed, several enhancements were identified that can be implemented in the future:

- **URL Shortener:** Add link analytics, custom aliases, and a public API for integration with other apps.
- **File Organizer:** Develop a graphical interface and support for organizing files in cloud storage (e.g., Google Drive).
- **Password Manager:** Integrate two-factor authentication (2FA), browser extensions, and cloud-based syncing with encryption.
- **Quiz Game:** Convert into a web-based quiz platform with user accounts, categories, and a leaderboard.

Implementing these features would further increase the usability, scalability, and real-world impact of the applications.

