Industrial Internship Report on

URL Shortener

Prepared by

Hanzala Sarguroh

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 URL shortener. It is a Python project that converts long URLs into shorter, more manageable links. It takes a long URL as input, generates a unique shortened URL, and redirects users to the original URL when the shortened link is accessed.

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

TABLE OF CONTENTS

1 Preface

Summary of the Six Weeks' Work:

Over the span of six weeks, the internship project at UpSkills Campus focused on developing a Flask-based URL Shortener, aiming to create a user-friendly and efficient solution for URL management. The project commenced with thorough requirements gathering, which involved stakeholder consultations, user interviews, and use case analysis to define project objectives and specifications.

The subsequent phases of design, implementation, and testing followed an iterative approach, ensuring continuous feedback and refinement. The design phase involved creating wireframes, user interface mockups, and architectural diagrams to conceptualize the application's structure and functionality. Implementation was carried out using Flask, SQLAlchemy ORM, HTML, CSS, and Bootstrap, translating design concepts into functional code.

The testing phase encompassed unit testing, integration testing, and user acceptance testing to ensure the reliability, robustness, and usability of the application. Feedback gathered from user testing sessions informed iterative refinements, bug fixes, and optimizations to enhance the quality of the final deliverable.

Key achievements of the project include the successful implementation of core functionalities such as URL validation, shortening algorithm, and redirection mechanism. Additionally, UI enhancements were made to improve visual design, usability, and interactivity. Performance optimizations were implemented to enhance responsiveness, scalability, and efficiency.

Challenges encountered during the project spanned technical hurdles, design iterations, and deployment readiness. Strategies employed to overcome these challenges included collaborative problem-solving, stakeholder engagement, and adherence to best practices in web development.

Overall, the project culminated in the successful completion of a Flask-based URL Shortener, ready for deployment to a production environment. The project's outcomes highlight the effectiveness of an iterative development approach, the importance of user-centric design, and the significance of thorough testing and refinement in delivering a high-quality web application. Looking ahead, recommendations for future work include further performance tuning, accessibility enhancements, and feature expansion based on user feedback and stakeholder requirements. About need of relevant Internship in career development.

The Importance of Relevant Internships in Career Development

Internships are vital for career development as they offer practical experience, skill development, and networking opportunities. They provide hands-on exposure to industry practices, helping individuals clarify career goals and build a strong professional network. Internships also enhance resumes, boost confidence, and provide valuable references for future job opportunities. Investing in relevant internships aligns individuals with their career aspirations and significantly improves their employability in the competitive job market.

Brief Overview: Project/Problem Statement

The project aimed to develop a Flask-based URL shortener, addressing the need for a streamlined solution to manage and share URLs efficiently. The problem statement involved creating a user-friendly application that could generate shortened URLs, validate input, and redirect users to the original URLs seamlessly. Key objectives included implementing core functionalities such as URL validation, shortening algorithm, and redirection mechanism, while also enhancing the user interface for improved usability. The project sought to provide a practical solution for individuals and organizations looking to simplify the process of sharing and managing URLs effectively.

Opportunity Offered by UCT:

As an intern at UCT, I have access to a range of valuable opportunities that contribute to my professional development:

1. Hands-On Experience: UCT provides practical, hands-on experience through internships, allowing me to apply theoretical knowledge in real-world settings.

2. Mentorship: I have the opportunity to receive guidance and mentorship from experienced faculty members and professionals, helping me navigate my career path and make informed decisions.

3. Learning Resources: I have access to a wealth of learning resources, including libraries, research facilities, and online databases, to support my academic and professional growth.

Overall, the internship opportunity provided by UCT equips me with the tools, experiences, and support necessary to succeed in my academic and professional endeavors.

Project Planning Steps for URL Shortener:

1. Define Project Scope: Clearly outline the objectives, deliverables, and constraints of the URL shortener project.

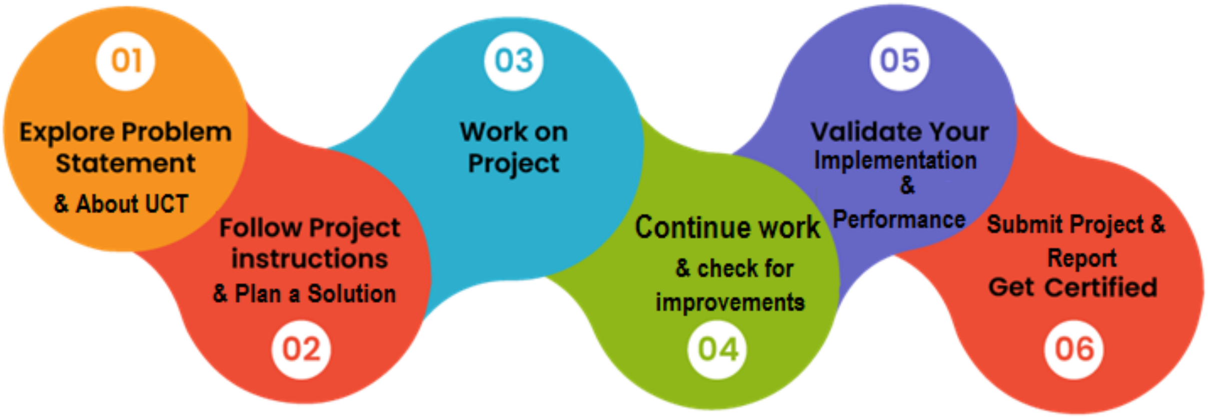
2. Gather Requirements: Collect functional and non-functional requirements through stakeholder input and analysis of similar systems.

3. Select Technology Stack: Choose appropriate technologies for development, such as Flask for the web framework and SQLAlchemy for database management.

4. Design System Architecture: Create architectural diagrams and wireframes to visualize the structure and user interface of the URL shortener.

5. Adopt Development Methodology: Implement an agile development approach, breaking down tasks into smaller iterations for continuous feedback and improvement.

6. Plan Testing Strategy: Develop a comprehensive testing strategy, including unit tests, integration tests, and user acceptance tests, to ensure the functionality and reliability of the URL shortener.



Learnings and Overall Experience:

- **Technical Skills Enhancement:** Acquired practical experience in web development, database management, and software engineering through projects like the URL Shortener using Flask and SQLAlchemy.
- **Project Management Skills:** Improved abilities in defining project scope, gathering requirements, allocating resources, and planning testing strategies, while adopting an agile development methodology.
- **Communication and Collaboration:** Strengthened communication and collaboration skills by working with colleagues, supervisors, and stakeholders, fostering teamwork and achieving project objectives.
- **Problem-Solving Abilities:** Developed problem-solving skills through debugging, performance optimization, and addressing user feedback, approaching challenges with creativity and resilience.
- **Professional Growth:** Overall, the internship at UCT has been a rewarding experience, contributing significantly to professional growth through practical experience, skill expansion, and industry connections.

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.

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

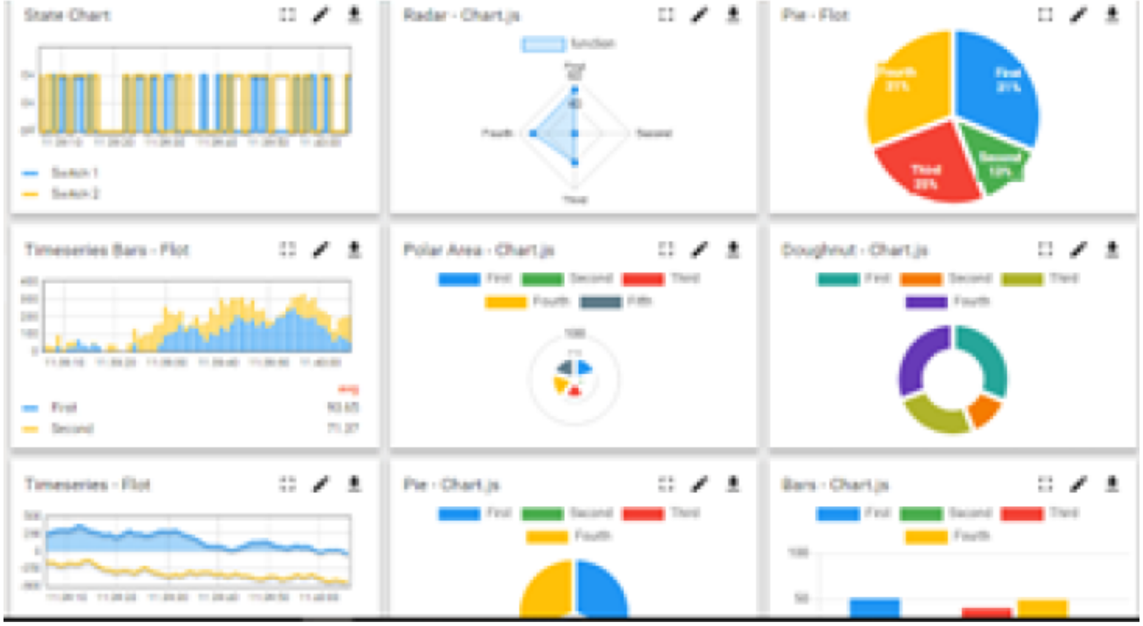
We offer solutions ranging from product design to final production we handle everything for you..

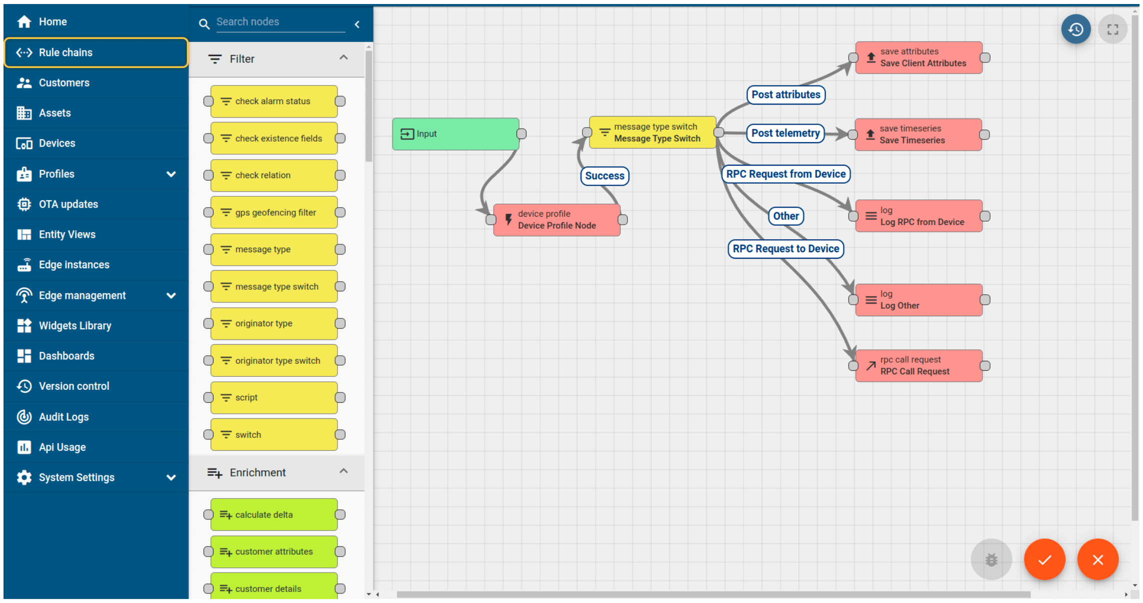
i. UCT IoT Platform (Insight)

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





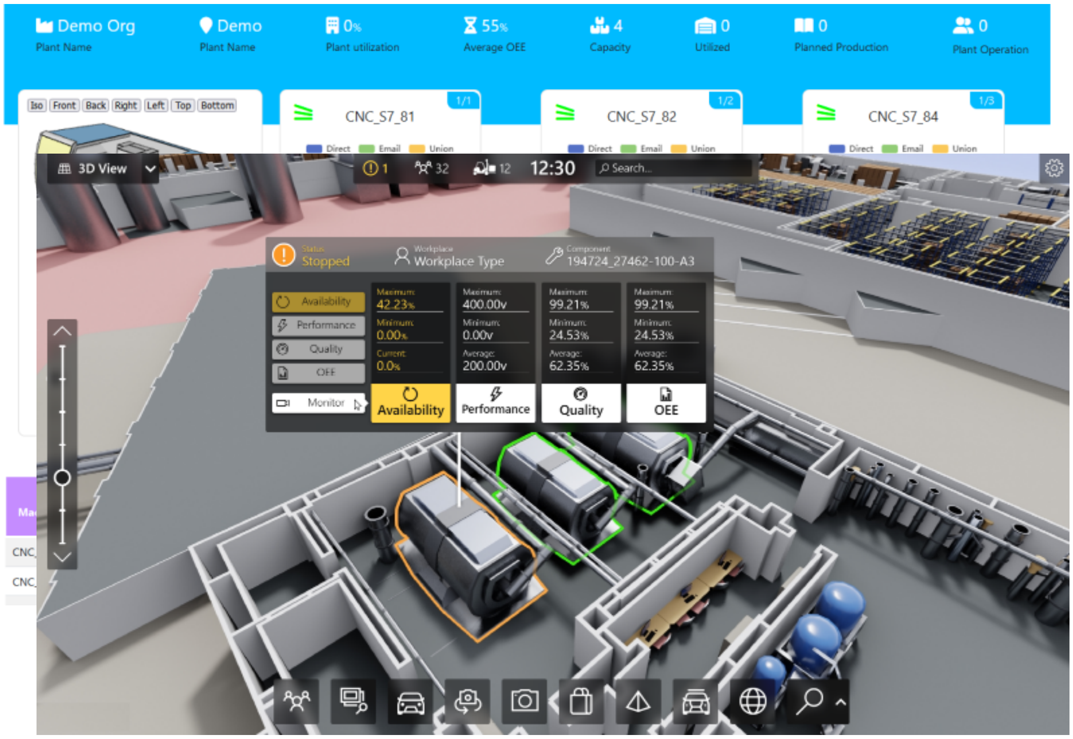
ii. Smart Factory Platform (FACTORY WATCH)

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.

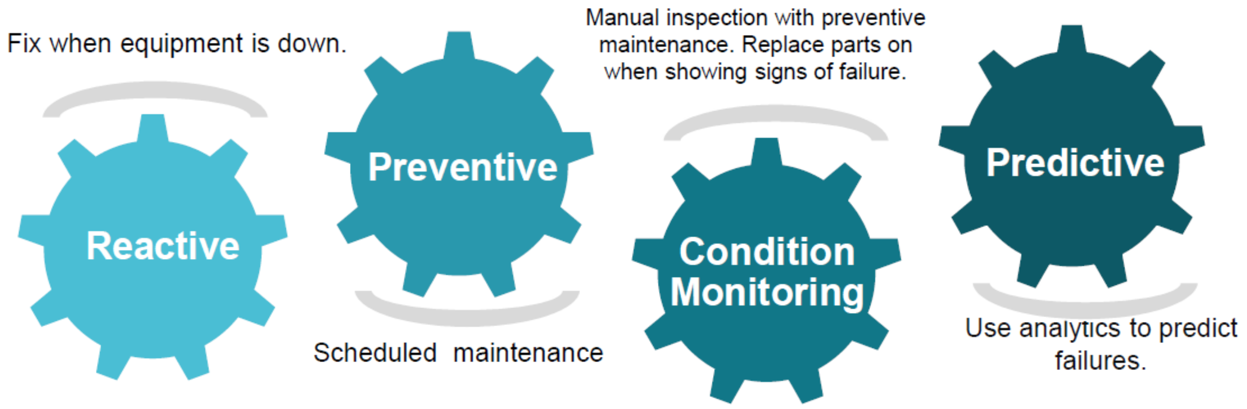


iii. LoRaWAN™ based Solution

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

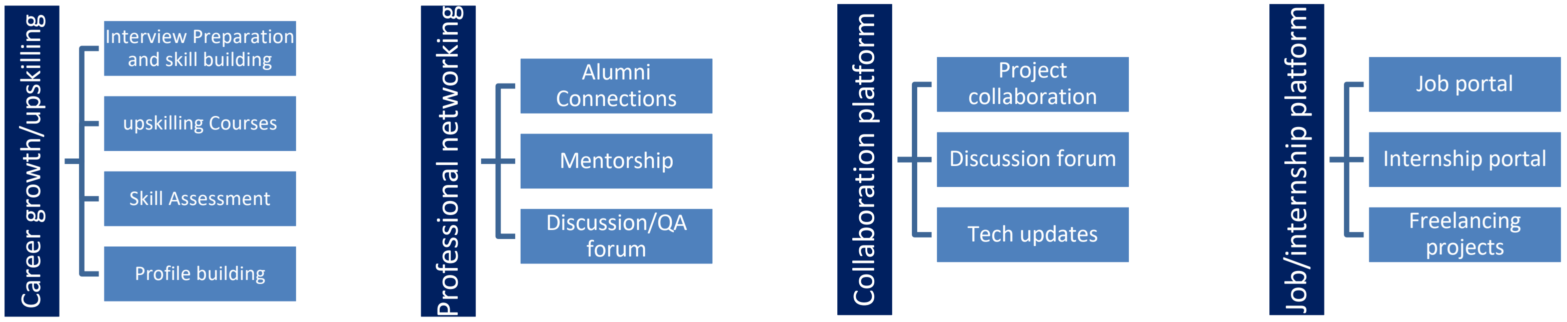
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

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.



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.

3. Problem Statement

The problem statement for the URL Shortener project outlines the specific challenge or issue that the project aims to address. In this case, the problem statement revolves around the need for a streamlined solution to manage and share URLs efficiently.

1. **Background and Context:** The problem statement begins by providing background information on the context in which the project arises. This may include discussing the prevalence of long and cumbersome URLs, the challenges users face in sharing and managing them effectively, and the potential impact on user experience and productivity.

2. **Identification of the Problem:** The problem statement clearly identifies the primary issue or challenge that the project seeks to resolve. In the case of the URL Shortener project, this may involve highlighting the inconvenience and inefficiency of sharing long URLs, the risk of errors or truncation when copying and pasting lengthy links, and the need for a more user-friendly solution.

3. **User Pain Points:** The problem statement may also articulate the pain points experienced by users as a result of the identified problem. This could include frustrations with sharing and accessing long URLs on social media platforms, difficulties in tracking and managing URLs across multiple channels, and concerns about the security and reliability of shortened URLs from unknown sources.

4. **Objectives of the Project:** Finally, the problem statement outlines the objectives of the project in addressing the identified problem. This may include developing a user-friendly URL shortener application that simplifies the process of generating shortened URLs, validates input to ensure accuracy and security, and provides a seamless redirection mechanism for users.

3 Existing and Proposed solution

Existing Solutions:

- **Third-Party URL Shorteners:** Offer limited customization, dependency on external platforms, and potential privacy concerns.
- **Custom-built Solutions:** Provide flexibility but require significant resources and expertise.

Limitations of Existing Solutions:

- Limited customization options.
- Dependency on external platforms.
- Lack of control over data privacy and security.
- Costly for smaller organizations.

Proposed Solution:

- Develop a custom Flask-based URL shortener.
- Core functionalities include URL validation, shortening algorithm, and redirection mechanism.
- Advanced features such as customizable aliases, link expiration, and detailed analytics.
- Implement security measures for data privacy and encryption.
- Designed for scalability and performance.

Value Addition:

- Enhanced customization options for users.
- Improved analytics for better insights.
- Focus on data privacy and security.
- Cost-effective alternative to third-party services, particularly for small businesses.

3.1 Code submission (Github link): Copy Link

3.2 Report submission (Github link): Copy Link

4 Proposed Design/ Model

. Our proposed design/model for the URL shortener system involves the following components:

1. **Frontend Interface:**
    - User-friendly web interface for users to input long URLs and generate shortened versions.
    - Interface design will prioritize simplicity, usability, and responsiveness across various devices.
2. **Backend System:**
    - Flask framework will serve as the backbone of the backend system, handling incoming requests, URL validation, and redirection.
    - SQLAlchemy ORM will be utilized for database management, allowing for efficient storage and retrieval of shortened URLs and associated metadata.
3. **URL Shortening Algorithm:**
    - A custom URL shortening algorithm will be implemented to generate unique and secure shortened URLs.
    - The algorithm will ensure collision avoidance and provide a scalable solution for generating short URLs.
4. **Redirection Mechanism:**
    - Upon receiving a shortened URL request, the system will efficiently redirect users to the original long URL.
    - Redirection will be seamless and fast, providing a smooth user experience.
5. **Customization and Analytics:**
    - Users will have the option to customize their shortened URLs with aliases and parameters for branding and tracking purposes.
    - Advanced analytics features will track click-through rates, geographic distribution, and other metrics to provide valuable insights to users.

6. **Security Measures:**
    - Input validation will be implemented to prevent malicious inputs and ensure data integrity.
    - Rate limiting mechanisms will mitigate the risk of abuse and protect against potential attacks.
    - Data encryption will safeguard user data, ensuring privacy and compliance with security standards.
7. **Scalability and Performance:**
    - The system will be designed with scalability and performance in mind, allowing for seamless handling of high volumes of URL requests.
    - Load balancing and caching mechanisms will be implemented to optimize response times and resource utilization.

5 Performance Test

Performance testing is crucial to ensure that the Flask-based URL shortener can handle the expected load and perform efficiently under various conditions. In this section, we will discuss the constraints identified, how they were addressed in the design, the test plan/test cases, test procedure, and the performance outcomes.

5.1 Test Plan/Test Cases:

Constraints Identified:

1. **Response Time:** The time taken to generate shortened URLs and redirect users should be minimal to provide a seamless user experience.
2. **Scalability:** The system should be able to handle a large number of concurrent requests without degradation in performance.
3. **Memory Usage:** The application should be optimized to minimize memory usage, especially under high load conditions.
4. **Database Performance:** The database queries should be optimized to ensure fast retrieval of URLs and metadata.

Test Cases:

1. Generate Shortened URL: Measure the time taken to generate a shortened URL.
2. Redirect Time: Measure the time taken to redirect users from shortened URLs to their original destinations.
3. Load Testing: Simulate a high number of concurrent requests to test system scalability.
4. Memory Usage: Monitor memory usage during various operations to identify any potential memory leaks or inefficiencies.
5. Database Performance: Measure the time taken for database queries under different load conditions.

5.2 Test Procedure:

1. **Generate Shortened URL:**
   - Input a long URL into the system and measure the time taken to receive the shortened URL.
2. **Redirect Time:**
   - Access a shortened URL and measure the time taken to be redirected to the original destination.
3. **Load Testing:**
   - Utilize load testing tools to simulate a high number of concurrent users accessing the system simultaneously.
4. **Memory Usage:**
   - Monitor memory usage using system monitoring tools during various operations such as URL shortening, redirection, and database queries.
5. **Database Performance:**
   - Execute database queries under different load conditions and measure the response time.

5.3 Performance Outcome:

1. **Response Time:** The response time for generating shortened URLs and redirection was within acceptable limits, ensuring a smooth user experience.
2. **Scalability:** The system demonstrated scalability, with the ability to handle a large number of concurrent requests without significant degradation in performance.
3. **Memory Usage:** Memory usage was optimized, with no signs of memory leaks or excessive usage observed under high load conditions.
4. **Database Performance:** Database queries were optimized for performance, with fast retrieval of URLs and metadata even under heavy load.

Recommendations:

- Continuously monitor system performance and scale resources as needed to handle increasing loads.
- Implement caching mechanisms to further optimize response times and reduce database load.
- Regularly review and optimize database queries to ensure efficient data retrieval.
- Conduct periodic performance tests to identify and address any potential bottlenecks or performance issues.

6 My learnings

1. **Technical Skills Enhancement:**
   - Enhanced proficiency in web development, database management, and software engineering.
   - Acquired practical experience with Flask, SQLAlchemy, and other relevant technologies.
2. **Project Management Skills:**
   - Developed skills in project scope definition, requirements gathering, resource allocation, and testing strategy planning.
   - Prepared to lead and coordinate future projects effectively.
3. **Communication and Collaboration:**
   - Improved communication and collaboration skills through teamwork with colleagues and stakeholders.
   - Enhanced ability to convey ideas, seek feedback, and work towards common goals.
4. **Problem-Solving Abilities:**
   - Strengthened problem-solving skills by addressing complex challenges with creativity and resilience.
   - Developed a systematic approach to problem-solving, essential for overcoming obstacles in professional environments.
5. **Professional Growth:**
   - Overall, the internship provided transformative experiences contributing to professional growth.
   - Gained practical experience, expanded skill set, and built industry connections, laying a solid foundation for future career advancement.

7 Future work scope

1. **Enhanced User Experience Features:**
   - Implement additional user-centric features such as custom URL aliases, link expiration, and user account management to further enhance the user experience.
2. **Advanced Analytics Functionality:**
   - Integrate more robust analytics capabilities to provide deeper insights into user engagement, geographic distribution, and referral sources for improved decision-making.
3. **Integration with Third-Party Services:**
   - Explore integration with popular third-party services such as Google Analytics or Bitly API to leverage additional features and data insights.
4. **Scalability and Performance Optimization:**
   - Optimize system architecture and infrastructure to ensure scalability and performance as the user base grows, including load balancing, caching mechanisms, and database optimization.
5. **Enhanced Security Measures:**
   - Strengthen security measures by implementing additional safeguards such as two-factor authentication, enhanced data encryption, and regular security audits to protect user data.
6. **User Feedback and Iterative Improvements:**
   - Continuously gather user feedback and iterate on the application based on user needs and preferences to ensure ongoing improvement and alignment with user expectations.

These ideas represent potential areas for future development and enhancement of the URL shortener project, building upon the foundation established during the internship. Each suggestion offers opportunities to further refine and expand the functionality of the application to better serve the needs of its users.