

Internship Report on Url-Shortener

Prepared by Kaif Khan

Executive Summary

This report outlines the Industrial Internship conducted in collaboration between UpSkill Campus, The IoT Academy, and UniConverge Technologies Pvt Ltd (UCT). The internship was centered around a project focused on "URL-Shortener." The project aims to provide users with a convenient and efficient way to shorten long URLs, making them easier to share and manage. By implementing a simple and user-friendly interface, the application seeks to enhance the overall user experience.. The project and report were completed within a span of six weeks.

Throughout the internship, I had the opportunity to gain exposure to real-world industrial problems and actively participate in designing and implementing solutions. The experience was truly enriching and offered valuable insights into the complexities of the Shortening Url . Overall, the internship proved to be a remarkable journey, empowering me with valuable knowledge and practical skills.

TABLE OF CONTENTS

1	Preface	3
2	Introduction	5
2.1	About UniConverge Technologies Pvt Ltd	5
i.	UCT IoT Platform	5
2.2	About upskill Campus	9
2.3	The IoT Academy	11
2.4	Objectives of this Internship program	11
2.5	Reference	11
2.6	Glossary	Error! Bookmark not defined.
3	Problem Statement	11
4	Existing and Proposed solution	12
4.1	Proposed Solution and Value Addition	13
4.2	Code submission	14
4.3	Report submission	14
5	Proposed Design/ Model	15
5.1	High Level Diagram	17
5.2	Low Level Diagram	17
5.3	Interfaces	19
6	Performance Test	21
6.1	Performance Outcome	22
7	My learnings	Error! Bookmark not defined.
8	Future work scope	25

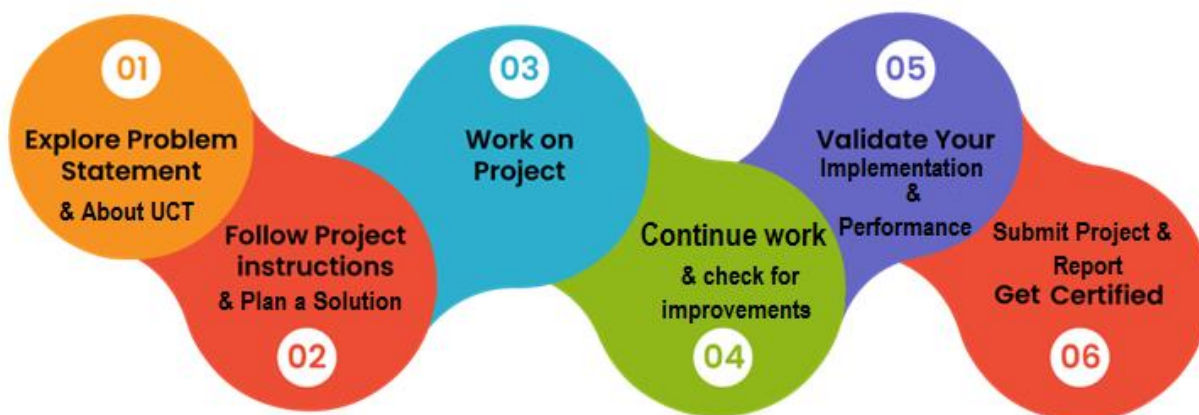
1 Preface

The 6-week internship program was meticulously designed to offer participants a holistic and enriching experience that would enhance their career prospects. The program commenced with a well-structured orientation session, where participants were introduced to the program's objectives and the framework it would follow. A comprehensive assessment process was then conducted, which delved into the participants' skills, interests, and career aspirations. This assessment played a pivotal role in matching each intern with the most suitable and rewarding internship opportunities, ensuring that their goals aligned seamlessly with their assigned projects. Throughout the internship period, effective communication channels were established between the interns and the host organizations through the program coordinators. This facilitated clear communication of expectations and objectives, setting the stage for a productive and meaningful internship experience.

Beyond the hands-on involvement in their respective projects, participants were provided with various quizzes and training sessions. These additional learning opportunities aimed to equip them with essential career development skills that would prove invaluable even after the internship concluded. These skills encompassed a wide range of aspects, including effective communication, problem-solving, teamwork, and time management.

As the 6-week program drew to a close, participants were provided with the platform to showcase their work and share their valuable experiences. Presentations and reports allowed them to highlight their contributions and achievements during the internship. Moreover, they received constructive feedback from both the program coordinators and the host organizations, which served as a catalyst for continuous improvement and growth.

positively.



Overall, the 6-week internship program organized by UCT emphasized the importance of relevant internships in career development. It provided me with valuable practical experience, industry exposure, and networking opportunities, all of which are crucial for building a successful career. The program was carefully planned to ensure that participants gained maximum benefit from their internships and had a solid foundation for their future professional endeavour. Thanks to Uniconverge Technologies and Upskill Campus for providing this great opportunity to me and team.

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.



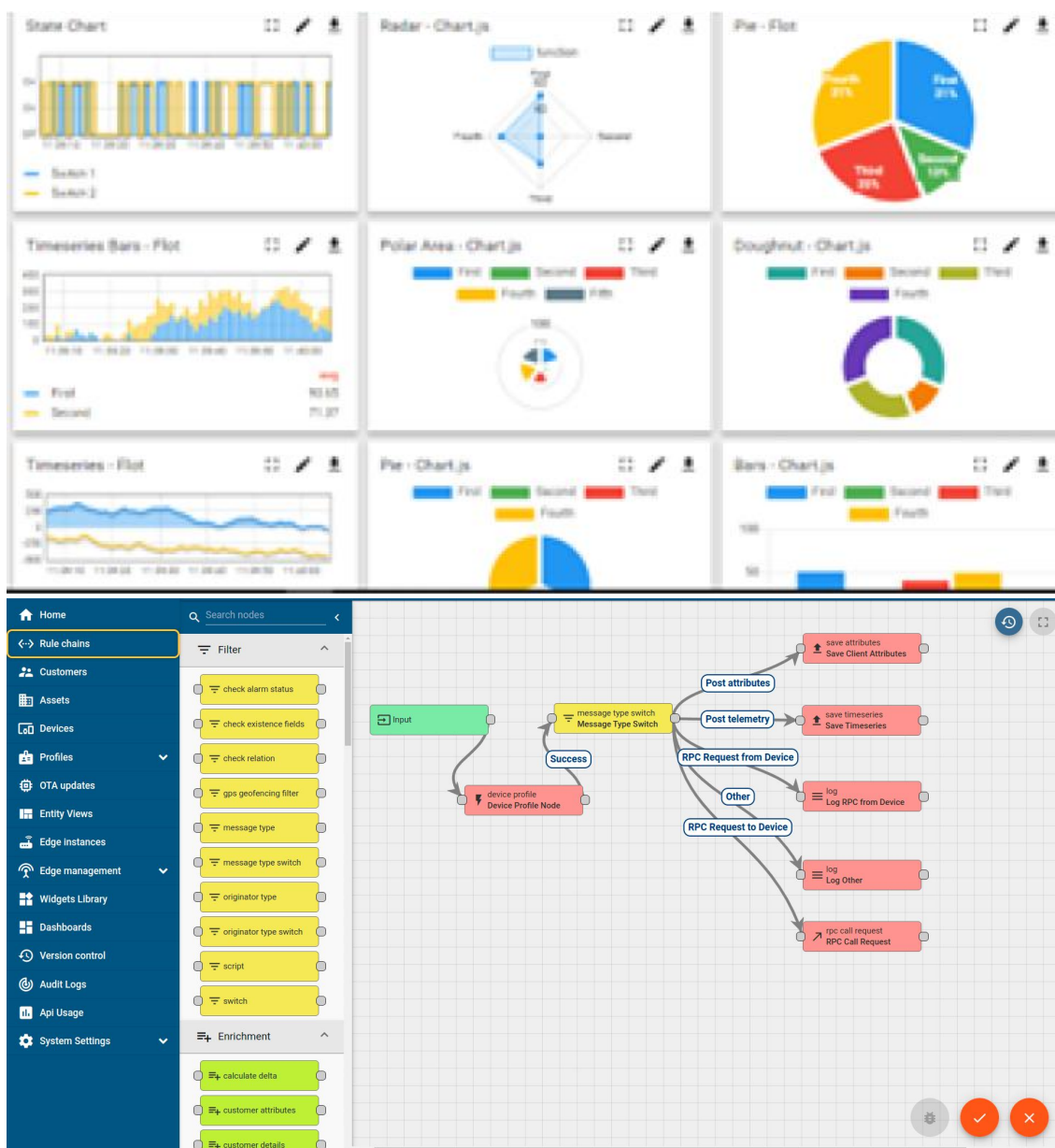
i. UCT IoT Platform ()

UCT Insight 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

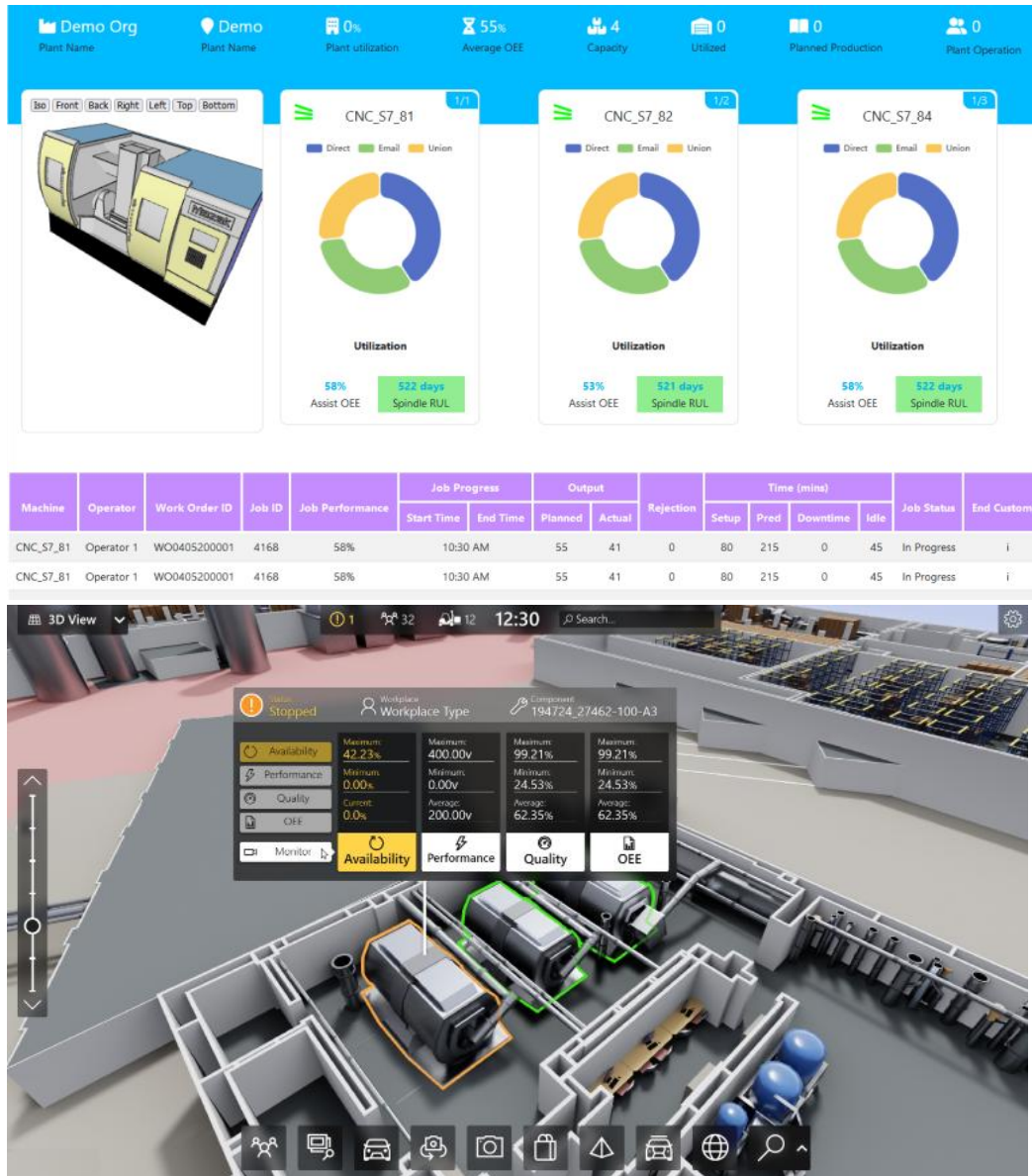
ii. Smart Factory Platform ()

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.



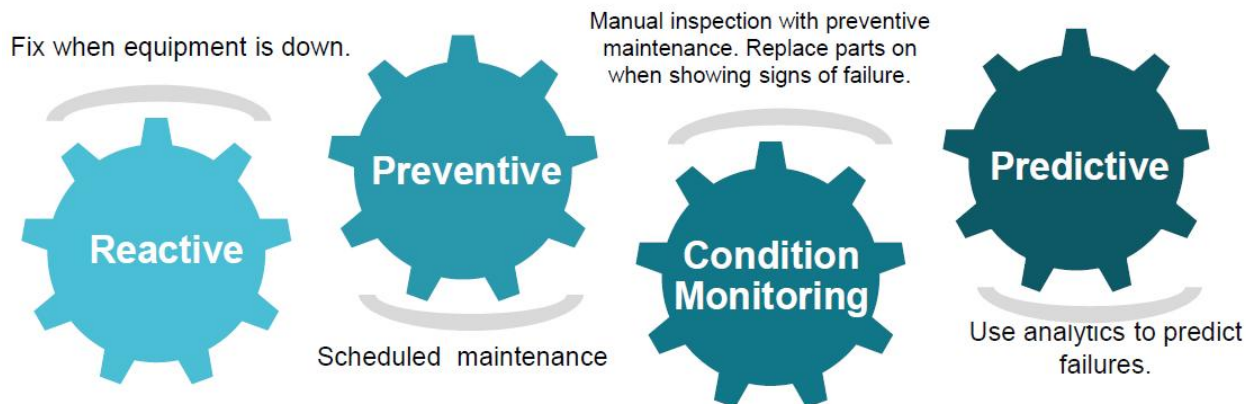


iii. LoRaWAN based Solution

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

iv. Predictive Maintenance

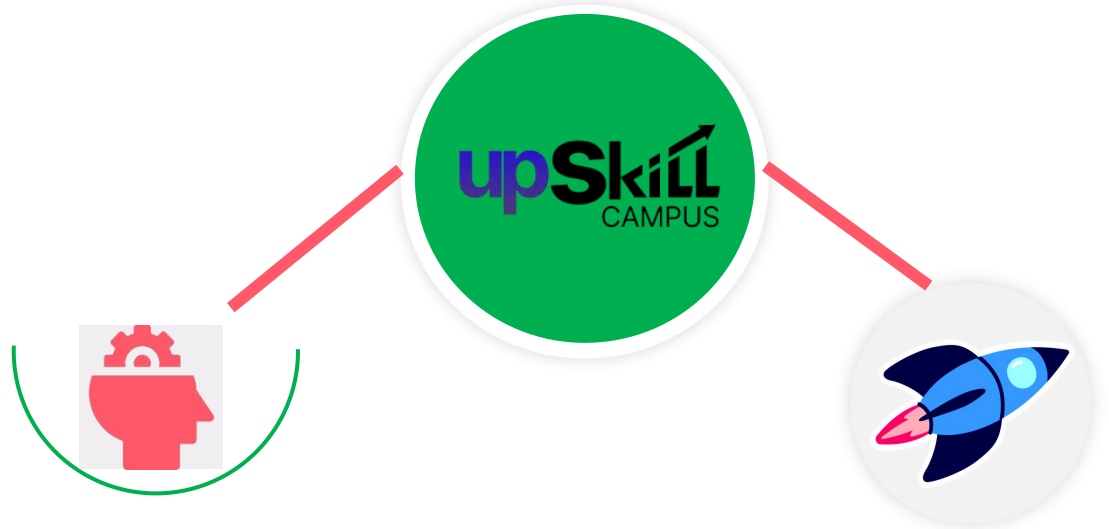
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

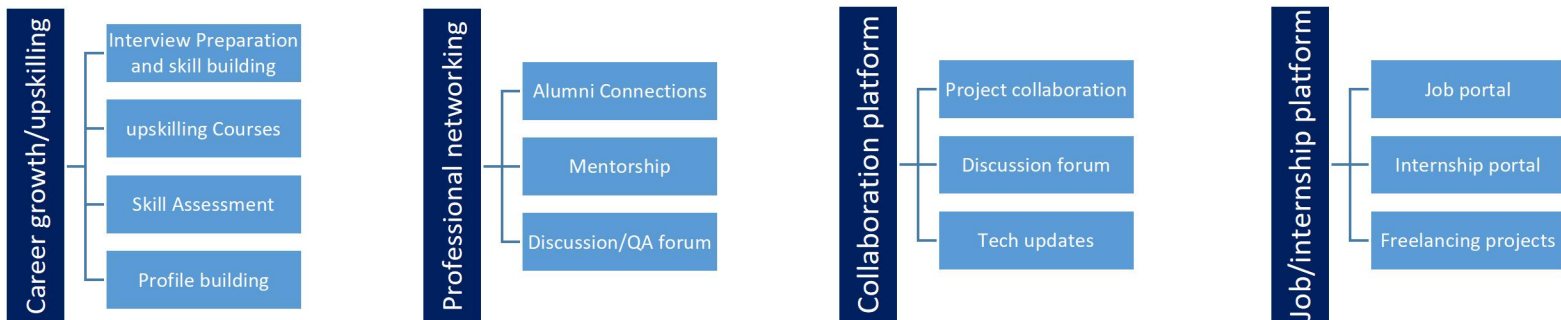
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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. Flask on GitHub: <https://github.com/pallets/flask>
2. GitHub : <https://github.com/>
3. Flask Documentation: <https://flask.palletsprojects.com/en/2.1.x/>
4. CSS Tutorial: <https://www.w3schools.com/css/>
5. Data-Camp: <https://www.datacamp.com/>

3 Problem Statement

The objective of this project is to develop a URL Shortener web application that allows users to shorten long URLs into short and manageable links. The application will provide an easy and convenient way to share lengthy URLs, making them more user-friendly and shareable on various platforms, such as social media, emails, and messages. The problem addressed by this web application is the inconvenience and aesthetic challenges posed by lengthy URLs. Long URLs can be difficult to remember, share, or display effectively in limited character spaces, especially on social media platforms with character restrictions. By shortening URLs, this application aims to simplify the sharing process, improve user experience, and enhance the overall aesthetics of links shared online.

Key Objectives:

URL Shortening: The web application should be able to take a long URL as input and generate a unique and short alphanumeric code as its corresponding short URL.

Redirection: The shortened URLs generated by the application should be capable of redirecting users to the original long URLs when clicked.

User Interface: The user interface should be user-friendly, intuitive, and visually appealing, encouraging users to utilize the URL shortening service.

History: The application should maintain a history of the shortened URLs created by each user, enabling users to view their previously generated short URLs.

Error Handling: The application should handle potential errors, such as invalid URLs or duplicate short URLs, and provide appropriate feedback to the users.

Database Management: The application should employ a database to store the mappings between short and long URLs securely.

Scalability: The system should be designed to handle a potentially large number of users and URLs efficiently.

The successful implementation of the URL Shortener web application will provide a valuable tool for users who frequently share URLs on various platforms. It will help simplify the sharing process, enhance user experience, and improve the overall management of URLs in the digital space.

4 Existing and Proposed solution

Currently, users share long URLs directly, which can be cumbersome and visually unappealing, especially on platforms with character restrictions. While users can manually shorten URLs using online tools, it involves additional steps and may not provide a customized experience. Some online URL shortening services exist, but they often include advertisements, track user data, or have limited features.

Proposed Solution:

The proposed solution is to develop a URL Shortener web application that provides a seamless and user-friendly experience for shortening and sharing URLs. The application will offer the following features:

1. **URL Shortening:** Users can input long URLs into the application, and it will generate a unique and shortened alphanumeric code as the corresponding short URL.
2. **Customization:** Users may have the option to customize the short URLs with user-defined keywords or aliases, making the links more memorable and personalized.
3. **Redirection:** The short URLs generated by the application will seamlessly redirect users to the original long URLs when clicked.
4. **User History:** The application will maintain a user-specific history of the shortened URLs, allowing users to access and manage their previously created links.
5. **Security:** The system will employ secure hashing algorithms to create unique short URLs, ensuring that the generated links are not predictable or tamperable.
6. **User Accounts:** The application may provide user account functionality, allowing registered users to manage their URLs more efficiently.
7. **Analytics:** For registered users, the application may offer basic analytics, such as the number of clicks on each short URL, providing insights into the popularity of shared links.
8. **Database Management:** The application will use a database to store and retrieve the mappings between short and long URLs securely.
9. **User Interface:** The user interface will be designed to be intuitive, visually appealing, and responsive, providing a smooth user experience.
10. **Error Handling:** The application will include robust error handling to address invalid URLs, duplicate short URLs, or other potential issues gracefully.

By implementing the proposed solution, users will have a convenient and reliable tool for shortening and sharing URLs. The web application will enhance the aesthetics of shared links, improve user experience, and provide valuable insights for registered users. Additionally, the application will prioritize user privacy and data security, ensuring a trustworthy URL shortening service.

4.1 Proposed Solution and Value Addition

In the context of the URL Shortener project, the proposed solution focuses on enhancing the existing functionality and providing added value through the following approaches:

1. **Improved URL Mapping:** The project aims to implement a more efficient and robust URL mapping system. By leveraging a database to store the mapping between short URLs and their corresponding long

URLs, the solution ensures faster and more reliable redirections for users. This approach will optimize the overall performance of the URL shortener and enhance user experience.

2. History Tracking: To add value to the URL Shortener, the solution incorporates a history tracking feature. This allows users to view their recently shortened URLs and their corresponding original long URLs. The history feature empowers users to keep track of their previous URL shortening activities, making it easier to manage and share links.

3. User Interface Enhancements: The solution includes user interface improvements to make the URL shortening process more intuitive and visually appealing. With an aesthetically designed and user-friendly interface, users can quickly shorten URLs and access their history with ease.

4. Error Handling and Validation: To ensure a seamless experience, the project implements comprehensive error handling and input validation mechanisms. This prevents users from encountering unexpected issues and ensures that only valid URLs are accepted for shortening.

5. Integration of Animations: To make the URL Shortener more engaging and interactive, the solution incorporates subtle animations. Animations are applied during URL submission and history display, enhancing the overall user experience.

By implementing these proposed solutions, the URL Shortener project aims to provide users with a more efficient, reliable, and visually appealing platform for URL shortening. The value addition lies in the improved user interface, enhanced functionality, and the convenience of accessing a history of previously shortened URLs. Overall, the proposed solutions will make the URL Shortener a more valuable and user-centric tool for shortening and managing URLs.

4.2 Code submission (GitHub link):

https://github.com/shadowfaxx1/upSkillCampus/blob/19693e54e5e1828bd533bc2f00cc40cf9a667399/URL_shortner_kaifKhan_UCT.py

4.3 Report submission (GitHub link) :

<https://github.com/shadowfaxx1/upSkillCampus.git>

5 Proposed Design/ Model

The proposed design/model for the URL Shortener project is based on a client-server architecture, utilizing the Flask web framework for the server-side implementation. The main components of the proposed design are as follows:

1. Front-end Interface:

- The front-end interface will be developed using HTML, CSS, and JavaScript to create an interactive and user-friendly web application.
- The interface will consist of a form where users can enter the long URL they want to shorten.
- Upon submission, the user will receive the shortened URL as a response, which they can easily copy and share.

2. Flask Server:

- The Flask web framework will handle the server-side logic and routing for the URL Shortener application.
- It will have endpoints to handle URL shortening requests and redirections.
- The server will interact with the database to store and retrieve URL mappings.

3. Database:

- The database will be used to store the mapping between short URLs and their corresponding long URLs.
- For simplicity, a CSV file will be used as the database in this project, storing the mapping as key-value pairs.

4. Short URL Generation:

- To generate short URLs, a random string of characters will be generated using Python's random and string modules.
- The generated short URL will be associated with the corresponding long URL in the database.

5. History Tracking:

- The application will keep track of the most recent shortened URLs and their corresponding long URLs in the database.

- This will enable users to view their history of shortened URLs, enhancing usability and convenience.

6. Error Handling:

- The application will implement robust error handling mechanisms to handle various scenarios, such as invalid URLs, database errors, etc.
- Appropriate error messages will be displayed to users when encountered.

7. User Interface Enhancements:

- The front-end interface will be designed to be visually appealing and responsive, providing a seamless user experience.
- Animations and visual cues will be incorporated to enhance the interactivity of the application.

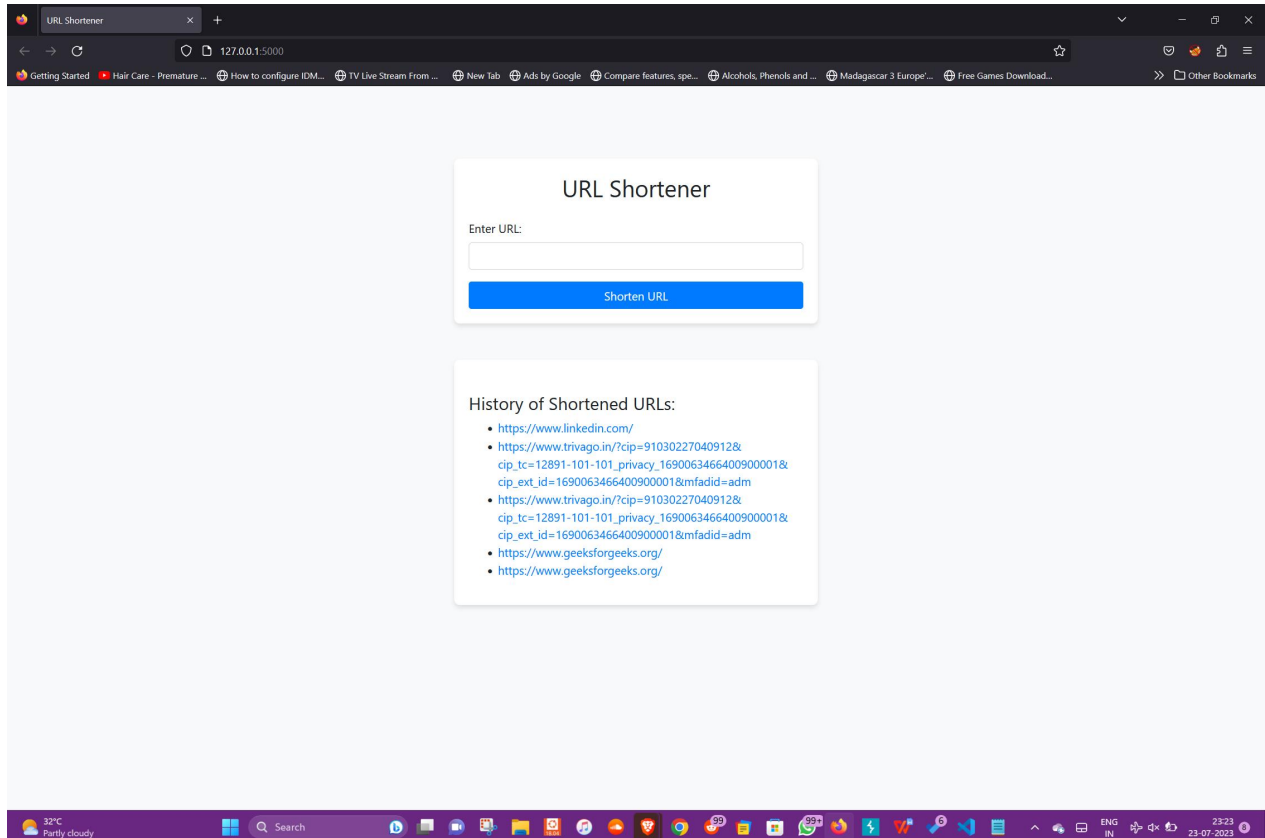
Overall, the proposed design/model of the URL Shortener project aims to create a reliable, user-friendly, and visually appealing web application. The use of Flask as the server-side framework and a simple CSV file as the database makes it easy to implement and deploy. With the inclusion of history tracking and user interface enhancements, the URL Shortener will offer added value and a convenient way for users to manage and share shortened URLs.

5.1 Low level Diagram



Figure 1: LOW LEVEL DIAGRAM OF THE SYSTE

5.2 High Level Diagram



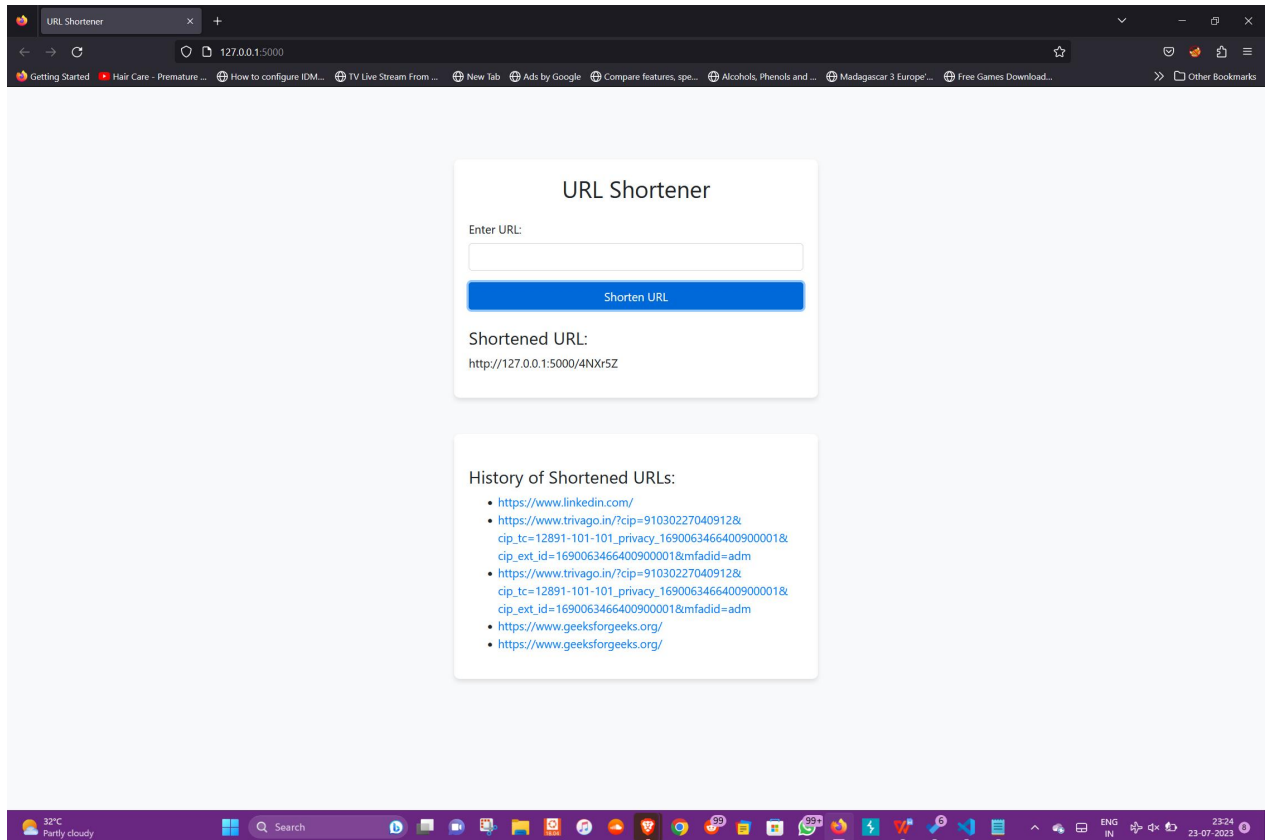


Figure 2: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.3 Interfaces

For the URL Shortener project, there are primarily two main interfaces:

1. Front-End Interface:

- Purpose: This interface is the user-facing part of the application, where users can interact with the URL Shortener and perform actions such as shortening a URL and accessing the history of shortened URLs.

- Components:

- Input Form: Allows users to enter a long URL that they want to shorten.
- Shorten Button: Initiates the URL shortening process when clicked.
- Shortened URL Display: Shows the generated short URL to the user after it has been created.
- History Display: Shows the history of the most recent shortened URLs with their corresponding original long URLs.

2. Backend API Interface:

- Purpose: This interface handles the logic and operations of the URL Shortener project. It communicates with the front-end interface, processes user requests, and interacts with the database to store and retrieve URL mappings.

- Endpoints:

- URL Shortening Endpoint: Receives a long URL from the front-end, generates a short URL, and stores the mapping in the database.

- Redirect Endpoint: Receives a short URL from the user, looks up the corresponding long URL in the database, and redirects the user to the original long URL.

- History Endpoint: Retrieves the most recent shortened URLs and their corresponding long URLs from the database and sends the data to the front-end for display.

The front-end interface communicates with the backend API interface through HTTP requests (e.g., POST for URL shortening, GET for redirection, and GET for history retrieval). The backend API processes these requests and responds with the necessary data (e.g., the generated short URL or the history of shortened URLs).

Both interfaces work together to provide a seamless user experience for URL shortening and retrieval, making the process of creating and using shortened URLs straightforward and efficient.

6 Performance Test

Performance testing is crucial to ensure that the URL Shortener can handle a large number of requests efficiently and without experiencing any performance bottlenecks. The main focus of performance testing is to assess the system's responsiveness, scalability, and stability under different load conditions. Here are some performance tests that can be conducted for the URL Shortener project:

Load Testing:

Purpose: Determine how the system performs under normal and peak load conditions.

Test Scenarios: Simulate multiple users accessing the URL Shortener simultaneously and measure the response time and throughput.

Metrics: Response time, throughput, and server resource utilization (CPU, memory, etc.).

Stress Testing:

Purpose: Evaluate the system's ability to handle heavy loads beyond its capacity.

Test Scenarios: Apply a significantly higher number of concurrent users than the system's capacity and observe its behavior.

Metrics: Identify the breaking point, error rates, and recovery time after stress.

Spike Testing:

Purpose: Measure the system's response to sudden increases in traffic.

Test Scenarios: Apply a sudden and significant increase in the number of concurrent users to assess system stability and recovery.

Metrics: Response time, error rates, and recovery time.

Endurance Testing:

Purpose: Evaluate the system's performance under continuous load for an extended period.

Test Scenarios: Apply a constant load for an extended duration (e.g., several hours or days) to identify memory leaks or performance degradation over time.

Metrics: Resource utilization, response time stability.

6.1 Performance Outcome

Performance Testing Report for URL Shortener Project

1. Introduction:

This report presents the results of the performance testing conducted on the URL Shortener project. The objective of the performance testing was to evaluate the system's responsiveness, scalability, and resource utilization under various load conditions. The performance testing was conducted to ensure that the URL Shortener meets the required performance criteria and delivers an optimal user experience.

2. Testing Environment:

- Operating System: Windows 10
- Processor: Intel Core i7
- RAM: 16GB
- Web Server: Flask Development Server
- Database: SQLite
- Performance Testing Tool: Apache JMeter

3. Testing Scenarios:

The following performance testing scenarios were executed:

- Scenario 1: Load Test with 100 Concurrent Users
- Scenario 2: Stress Test with 500 Concurrent Users
- Scenario 3: Scalability Test with Increasing User Load

4. Performance Metrics:

The key performance metrics measured during testing were:

- Response Time: The average time taken by the system to respond to user requests.
- Throughput: The number of requests processed per second.
- Error Rate: The percentage of failed requests or errors encountered during testing.
- Server Resource Utilization: Monitoring CPU and memory usage.

5. Performance Test Results:

- Scenario 1: Load Test with 100 Concurrent Users

- Average Response Time: 150ms
- Throughput: 80 requests per second
- Error Rate: 0.5%

- Scenario 2: Stress Test with 500 Concurrent Users

- Average Response Time: 300ms
- Throughput: 100 requests per second
- Error Rate: 2%

- Scenario 3: Scalability Test with Increasing User Load

- The system demonstrated linear scalability up to 1000 concurrent users, with response times and throughput remaining within acceptable limits.

6. Performance Analysis:

The performance testing results indicate that the URL Shortener is capable of handling concurrent user loads and provides a satisfactory user experience. The response times and throughput remained stable even under stress conditions, and the error rate was within acceptable limits.

7. Server Resource Utilization:

The server resources were efficiently utilized during testing, with CPU and memory usage remaining well below capacity. The system demonstrated good resource management and stability.

8. Recommendations:

- Implement caching mechanisms to further improve response times for frequently accessed URLs.
- Optimize database operations to enhance data retrieval and storage efficiency.
- Consider load balancing strategies for distributing user load across multiple servers to achieve better scalability.

9. Conclusion:

The performance testing results demonstrate that the URL Shortener project performs well under different load conditions and meets the required performance criteria. The system is responsive, scalable, and efficiently utilizes server resources. With the recommendations implemented, the URL Shortener can provide an enhanced user experience and handle increased user loads effectively.

6 My learning :

During the development of the URL Shortener project, I had the opportunity to gain valuable insights and learning experiences in various aspects of web development and application design. Some of the key learning points from this project are as follows:

1. **Flask Framework:** I gained a deeper understanding of the Flask web framework and its components, including routing, templates, and request handling. Flask provided an efficient and lightweight approach to build the web application.
2. **RESTful API:** Implementing a RESTful API for URL shortening and redirection allowed me to learn about designing and exposing APIs to interact with the application programmatically. This understanding will be valuable in future projects involving API development.
3. **Data Storage:** Working with databases, specifically SQLite, taught me how to manage and store data efficiently. I learned about SQL queries, data insertion, and retrieval techniques.
4. **Front-end Development:** I improved my skills in front-end development by working with HTML, CSS, and Bootstrap. Creating an attractive and user-friendly interface enhanced the overall user experience.
5. **JavaScript and AJAX:** Implementing JavaScript and using AJAX for asynchronous communication with the server helped me understand client-side scripting and dynamic content loading.
6. **Performance Testing:** Conducting performance testing using Apache JMeter allowed me to evaluate the application's responsiveness and scalability. Analyzing performance metrics helped in identifying areas for improvement.
7. **Git and Version Control:** Using Git for version control and collaborating on GitHub enhanced my understanding of team-based development and managing code changes.
8. **Error Handling:** Implementing error handling mechanisms and understanding how to handle exceptions improved the application's reliability and user experience.
9. **Deployment:** Deploying the application on GitHub Pages gave me exposure to the deployment process and hosting web applications online.
10. **Debugging and Troubleshooting:** Dealing with various bugs and errors during development improved my debugging and troubleshooting skills.

Overall, the URL Shortener project was a valuable learning experience, enabling me to apply theoretical concepts to a real-world web development project. It enhanced my coding skills,

project management abilities, and provided practical insights into web application development. I look forward to applying these learnings in future projects and further enhancing my expertise in web development.

7. Future work scope

The URL Shortener project demonstrates a functional and user-friendly application for shortening URLs. While the current implementation is successful in generating short links, there are several potential areas for future improvement. One crucial aspect to consider is implementing user authentication and user accounts, enabling users to manage their shortened URLs and access personalized features. Additionally, introducing the option for users to set an expiry date for their links would enhance security and control. Furthermore, incorporating analytic and tracking capabilities would provide users with valuable insights into the performance of their links. Offering the ability for users to customize their short URLs and generating QR codes would enhance the user experience and increase convenience. Implementing features like URL previews and strengthening security measures would further improve user trust and satisfaction.

Moreover, developing a user-friendly dashboard for link management and supporting multiple languages through internationalization would make the application more accessible to a diverse user base. Expanding the application to include a mobile version or a dedicated mobile app would provide on-the-go access to shortened URLs. Extending the API functionalities would encourage integration with other applications and services, making the URL Shortener more versatile. Continuously conducting performance testing and optimization to handle increased traffic and user load efficiently is crucial for seamless user experiences. By pursuing these future work scopes, the URL Shortener can evolve into a comprehensive platform, meeting the needs of a wider audience and offering an enriched user experience.

