

Industrial Internship Report on

URL Shortener

Prepared by

Bhojraj Lalit Narware

Executive Summary

This report offers comprehensive insights into the Industrial Internship facilitated by upskill Campus and The IoT Academy, in collaboration with esteemed Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

The primary focus of this internship was centered around a specific project/problem statement presented by UCT. The allotted time frame for project completion, including the submission of this report, was limited to a period of 6 weeks.

My assigned project revolved around the development of a URL shortener tool, which entailed the creation of a utility capable of converting lengthy URLs into concise and more manageable versions. The scope of the project encompassed various backend development tasks, such as the implementation of an efficient URL shortening algorithm. Additionally, integration of a database for storing the generated short URLs was a crucial aspect of the project. The establishment of API endpoints to handle both shortening requests and the subsequent redirection was an integral part of the project's objectives. While not mandatory, the incorporation of frontend development elements, including a user interface and robust input validation, was encouraged to elevate the overall user experience.

TABLE OF CONTENTS

1. Preface	3
2. Introduction	11
2.1 About UniConverge Technologies Pvt Ltd	8
2.2 About upskill Campus	11
2.3 The IOT Academy	15
2.4 Objective	15
2.5 Reference	17
2.6 Glossary	198
3. Problem Statement	20
4. Existing and Proposed solution	22
5. Proposed Design/ Model.....	26
5.1 High Level Diagram (if applicable)	277
6. Performance Test	28
6.1 Test Plan/ Test Cases	29
6.2 Test Procedure	Error! Bookmark not defined.3
7. My learnings.....	35
8. Future work scope	37

1 Preface

Summary of the whole 6 weeks' work:

Over the course of six engaging weeks, my industrial internship provided a transformative learning experience in Python programming and web development. Collaboratively organized by Upskill Campus, The IoT Academy, and esteemed industrial partner UniConverge Technologies Pvt Ltd (UCT), the internship centered around the development of a URL shortener project, which presented an exciting opportunity to apply theoretical knowledge in a practical context.

The internship commenced with an insightful orientation, where the project's scope and objectives were delineated. The project's focal point was the creation of a URL shortener tool, designed to convert lengthy web addresses into concise and manageable versions. This endeavor necessitated a comprehensive understanding of backend development principles, database integration, and API design.

The initial weeks were dedicated to an immersive Python programming journey. Through a series of hands-on workshops and guided exercises, I honed my skills in Python, mastering its syntax, data structures, and object-oriented programming concepts. These foundational skills were crucial for the subsequent stages of the project.

With a solid Python foundation, I delved into the intricate world of web development. Leveraging Python's powerful frameworks and libraries, I embarked on designing and implementing the URL shortener project. This involved the meticulous development of a URL shortening algorithm, optimizing its efficiency and reliability. The integration of a database provided the means to store and retrieve the generated short URLs, fostering a seamless user experience.

The heart of the project lay in crafting robust API endpoints. These endpoints seamlessly managed incoming shortening requests and directed users to the appropriate original URLs, showcasing the

dynamic interplay between frontend and backend development. While optional, I recognized the value of frontend development in enhancing user engagement. Thus, I dedicated efforts to create an intuitive user interface, employing HTML, CSS, and JavaScript to ensure a cohesive and user-friendly platform.

As the project's completion neared, the focus shifted to deployment and security considerations. The application was successfully hosted on a web server, bolstered by comprehensive security measures to safeguard user data and system integrity.

In conclusion, this six-week internship was an exhilarating journey through Python programming and web development realms, culminating in the creation of a functional and impactful URL shortener tool. The experience not only solidified my technical skills but also instilled a profound appreciation for problem-solving in real-world scenarios. The collaborative environment fostered by upskill Campus, The IoT Academy, and UCT enriched my learning, making this internship an invaluable stepping stone in my professional journey.

About need of relevant Internship in career development:

The need for relevant internships in career development cannot be overstated. Internships play a pivotal role in equipping individuals with practical skills, industry insights, and valuable experiences that significantly enhance their professional growth. Here are some key reasons why relevant internships are essential for career development:

1. Hands-on Learning: Internships provide a platform for individuals to apply theoretical knowledge gained in academic settings to real-world scenarios. This hands-on experience bridges the gap between theory and practice, allowing interns to develop practical skills that are vital for success in their chosen fields.

2. Industry Exposure: Internships offer a unique opportunity to gain firsthand exposure to the workings of a specific industry or sector. This exposure helps individuals understand industry trends, challenges, and best practices, enabling them to make informed career decisions and align their skill development accordingly.

3. Networking Opportunities: Internships allow individuals to connect with professionals, mentors, and peers within their chosen industry. Building a strong professional network during an internship can open doors to future job opportunities, collaborations, and mentorship relationships.

4. Skill Development: Internships provide a platform to acquire and refine a wide range of skills, both technical and soft. These skills can include project management, communication, teamwork, problem-solving, and adaptability – all of which are highly valuable in any career.

5. Resume Enhancement: A relevant internship adds substantial weight to an individual's resume. It demonstrates to potential employers that the individual has practical experience and a clear understanding of the demands of the industry.

6. Career Clarity: Internships offer a chance to explore different roles and industries, helping individuals gain clarity about their career preferences and goals. This firsthand experience can guide them in making informed decisions about their future career path.

7. Professional Development: Internships provide opportunities for professional development through training sessions, workshops, and exposure to real-world challenges. This learning experience contributes to an individual's personal growth and overall career trajectory.

8. References and Recommendations: A successful internship can lead to strong references and recommendations from supervisors or mentors. Positive feedback from professionals in the industry can bolster an individual's credibility and increase their chances of securing desirable job positions.

9. Job Placement: Many organizations consider their intern pool when hiring for full-time positions. A well-performed internship can lead to a job offer, allowing interns to seamlessly transition into their desired careers.

10. Confidence Building: Successfully completing an internship instills confidence in an individual's abilities, confirming that they can contribute effectively in a professional setting. This newfound confidence can positively impact job interviews and performance in future roles.

In essence, relevant internships serve as a bridge between academic learning and professional success. They provide a dynamic learning environment where individuals can grow, learn, and make meaningful contributions, setting the stage for a rewarding and fulfilling career journey.

Brief about Your project/problem statement:

My project/problem statement revolves around creating a URL shortener tool, designed to streamline and simplify lengthy web addresses. The objective is to develop a robust and efficient system that takes long URLs as input and generates shorter, more manageable versions that redirect users to the original content.

The key components of the project include:

1. URL Shortening Algorithm: The heart of the project lies in designing and implementing an effective URL shortening algorithm. This algorithm should generate unique and concise short URLs from the provided long URLs. The efficiency of this algorithm is crucial to ensure quick redirection and minimize any potential conflicts.

2. Database Integration: To maintain the association between the original long URLs and their corresponding short versions, a database will be integrated into the system. This database will store pairs of long and short URLs, allowing for efficient retrieval and redirection.

3. API Development: The project involves creating API endpoints that handle incoming requests for URL shortening and redirection. These APIs will communicate with the URL shortening algorithm and the database to facilitate the conversion and redirection processes seamlessly.

4. Frontend Interface (Optional): While not mandatory, developing a user-friendly frontend interface can enhance the user experience. This interface may allow users to input long URLs and receive their corresponding short versions, along with convenient features like copy-to-clipboard functionality.

5. Deployment and Security: Once the core functionality is established, the application will be deployed on a web server. Security measures will be implemented to safeguard user data, prevent unauthorized access, and ensure the integrity of the system.

The project's significance lies in its practicality and relevance to web development and data management. By tackling the challenges of creating a functional and user-friendly URL shortener, the project provides an opportunity to apply concepts from various domains, including backend development, database management, API design, and potentially frontend development. This hands-on experience equips participants with valuable skills for solving real-world problems and contributes to their growth as aspiring professionals in the field of web development.

Opportunity given by USC/UCT:

Upskill Campus (USC) or The IoT Academy in collaboration with UniConverge Technologies Pvt Ltd (UCT) as of my knowledge cutoff in September 2021, internships and collaborations with industry partners generally offer valuable opportunities for professional growth and development.

Participating in an internship or collaboration program with USC/UCT could provide the following opportunities:

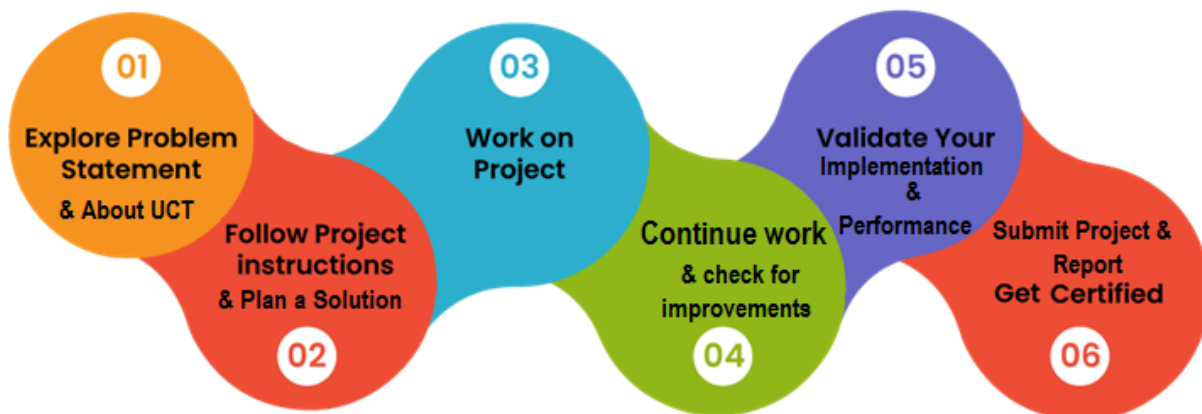
1. Industry Exposure: Working with a renowned institution and industry partner gives you exposure to real-world industrial problems, challenges, and practices. It allows you to gain insights into the operations and dynamics of the industry, which can be invaluable for your career.

2. Practical Experience: Internships and collaborations offer hands-on experience, enabling you to apply your knowledge and skills in a practical setting. Working on projects or problem statements provided by industry partners helps bridge the gap between theory and practice, enhancing your understanding and proficiency in your chosen field.

3. Networking: Collaborating with USC, UCT, and their industry partner gives you the opportunity to build a professional network. You can connect with professionals, mentors, and experts in the industry, which can lead to future job prospects, references, and valuable connections.

4. Skill Development: Engaging in projects and problem-solving within the context of an internship or collaboration provides an avenue for skill development.

How Program was planned:



Learnings and overall experience:

Learnings and overall experience you might expect from working on a URL shortener project during a Python internship.

1. Technical Skills: Working on a URL shortener project in Python allows you to enhance your technical skills in various areas. You'll gain proficiency in Python programming.

2. Problem-Solving: Developing a URL shortener involves addressing specific challenges, such as generating unique shortened URLs, managing the database, and implementing the redirection mechanism. Throughout the project, you'll enhance your problem-solving skills by identifying and implementing effective solutions to these challenges.

3. Project Management: Participating in an internship project like a URL shortener involves managing your time and tasks effectively. You'll gain experience in project planning, task prioritization, and meeting deadlines. Additionally, you may work collaboratively with a team or receive guidance from mentors, providing an opportunity to improve your communication and teamwork skills.

4. Debugging and Testing: During the implementation phase, you'll encounter and overcome challenges through debugging and testing. This process will enhance your troubleshooting skills and help you develop strategies for identifying and resolving issues in your code.

5. Exposure to Industry Practices: Internships often provide exposure to industry practices and standards. You may learn about coding conventions, version control systems (such as Git), code documentation, and collaborative development workflows. These experiences will help you align your skills and practices with industry expectations.

Engaging in a Python internship centered around a URL shortener project presents a unique chance to put your programming acumen into action within a tangible, real-life context. This endeavor facilitates the honing of technical prowess, the cultivation of effective problem-solving skills, the

development of adept project management abilities, and an immersion in industry practices. It is advised to proactively seek guidance from mentors, actively collaborate with fellow interns, and maximize the wealth of knowledge and skills attainable during this immersive learning journey.

Gratitude extends to Upskill Campus (USC) and The IoT Academy for orchestrating this enriching internship, in conjunction with the esteemed collaboration of UniConverge Technologies Pvt Ltd (UCT). Their collective efforts have provided an invaluable platform for practical learning and professional growth, underscoring the importance of hands-on experiences in shaping a well-rounded and capable individual in the realm of Python programming and web development.

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.

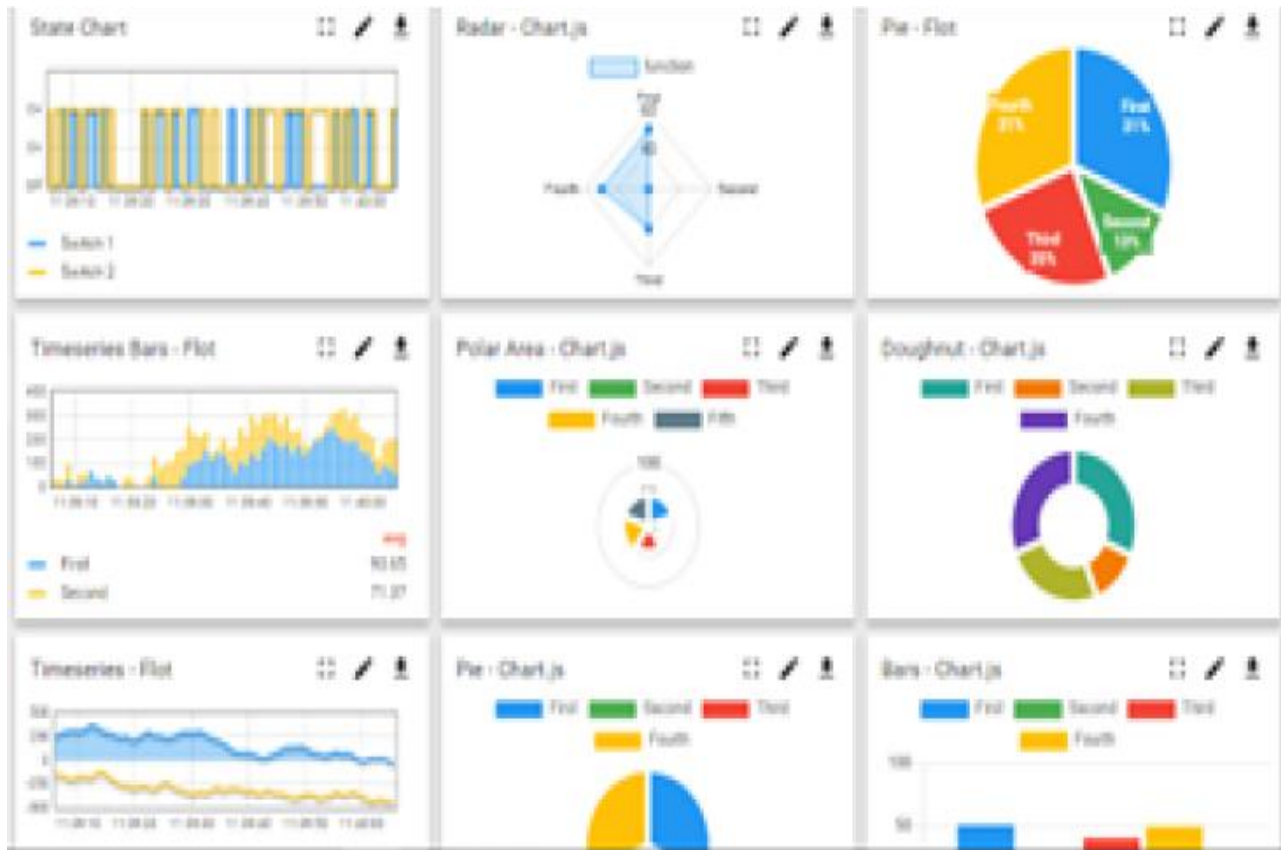
For developing its products and solutions it is leveraging various **Cutting Edge Technologies** e.g. **Internet of Things (IoT)**, **Cyber Security**, **Cloud computing (AWS, Azure)**, **Machine Learning**, **Communication Technologies (4G/5G/LoRaWAN)**, **Java Full Stack**, **Python**, **Front end**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

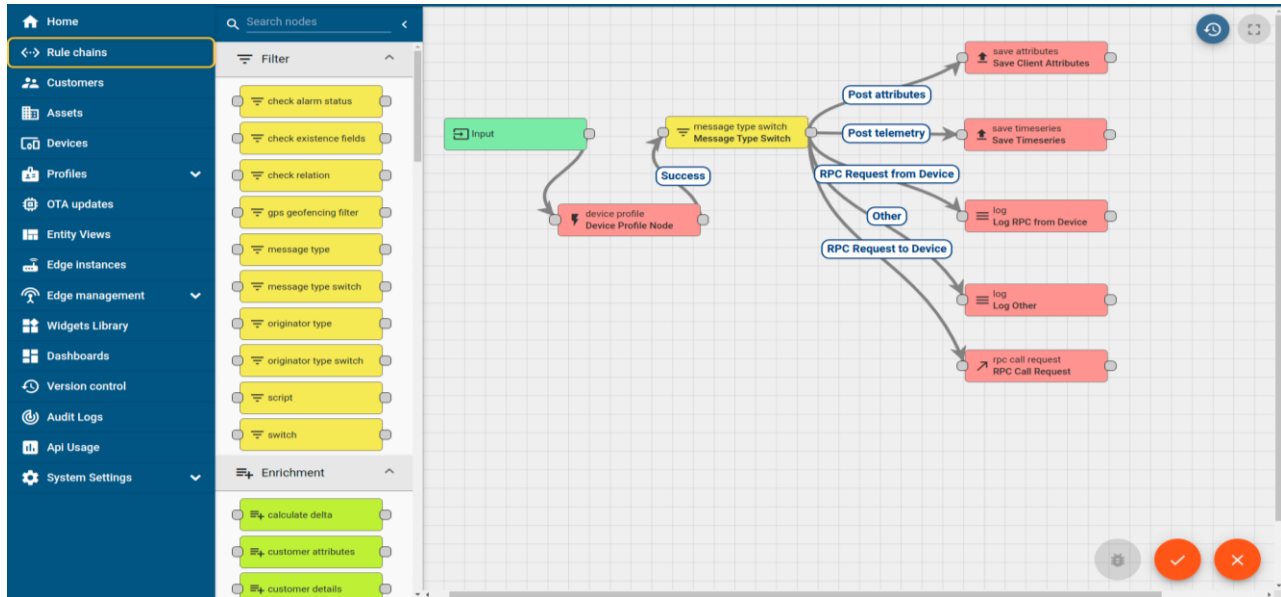


i. UCT IoT Platform ()

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.





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 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.

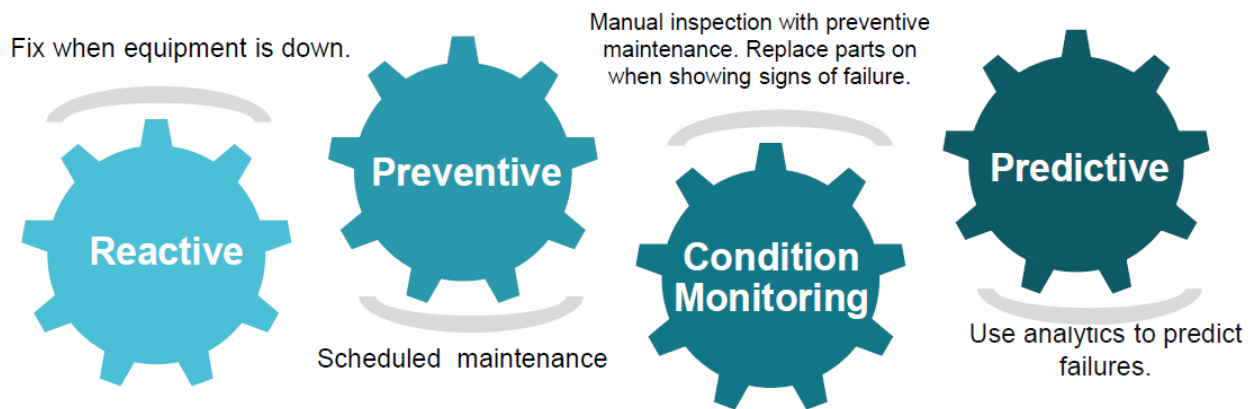


iii. based Solution

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

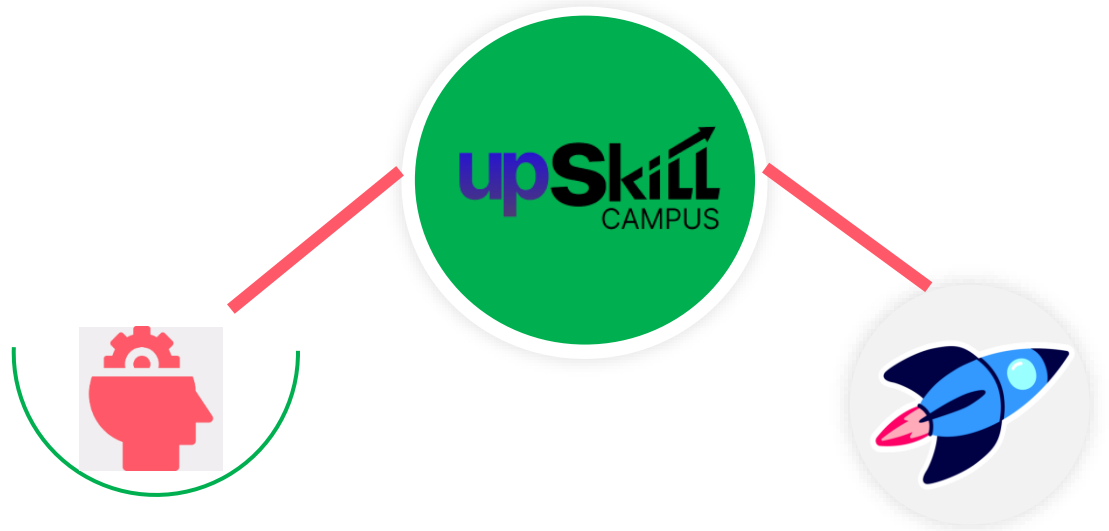
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

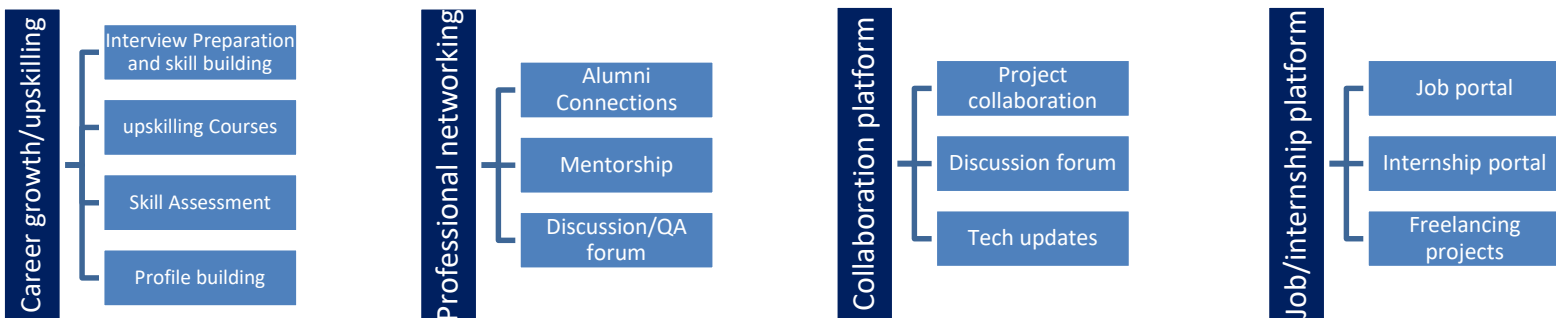
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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

Certainly, here are the shortened objectives of this internship program:

- 1. Practical Skill Application:** Apply classroom learning to real-world projects like the URL shortener.
- 2. Enhance Technical Proficiency:** Develop skills in Python, web development, APIs, and databases.
- 3. Boost Problem-Solving Skills:** Tackle challenges like efficient algorithms and secure data storage.
- 4. Learn Project Management:** Adhere to timelines, prioritize tasks, and achieve milestones.
- 5. Gain Industry Insights:** Understand industry practices, trends, and dynamics through collaboration with UCT.
- 6. Explore Various Disciplines:** Experience backend, frontend (optional), and database aspects of web development.
- 7. Refine Communication:** Improve communication with mentors, peers, and potential users.
- 8. Build Networks:** Establish connections for mentorship, references, and future opportunities.
- 9. Personal Growth:** Overcome obstacles, enhance confidence, and achieve tangible results.
- 10. Deliver Tangible Project:** Successfully complete the URL shortener project for resume and portfolio enhancement.

In summary, the internship program aims to provide practical experience, technical growth, industry insights, and personal development to prepare participants for a successful career in Python programming and web development.

2.5 Reference

[1] **Bitly API:** Bitly is a popular URL shortening service that provides an API for developers to integrate shortening and analytics features into their applications. You can find more information about their API documentation here:

<https://dev.bitly.com/docs/getting-started/authentication/>

[2] **TinyURL:** TinyURL is another well-known URL shortening service. Although they don't provide an official API, you can refer to their website and examine how they generate and redirect shortened URLs:

<https://tinyurl.com/>

[3] **Open source URL Shortener Projects:** If you're interested in exploring open-source implementations of URL shorteners, you can check out projects like YOURLS (<https://yourls.org/>) (<https://polrproject.org/>).

2.6 Glossary

Terms	Acronym
URL	Uniform Resource Locator - A web address that specifies the location of a resource on the internet.
Shortened URL	A compressed or abbreviated version of a URL that redirects to the original, longer URL when accessed.
Redirect	The process of automatically forwarding a user from one URL to another. In the context of URL shorteners, clicking on a shortened URL triggers a redirect to the original long URL
API	Application Programming Interface - A set of rules and protocols that allows different software applications to communicate with each other. APIs are often used in URL shorteners to handle URL shortening requests and redirection.
Backend	The server-side of a web application that processes requests, manages data, and handles the logic behind the scenes. In a URL shortener project, the backend code would handle generating shortened URLs, storing them, and managing the redirection process.

3 Problem Statement:

Design and develop a URL shortener application using Tkinter, a Python GUI library. The application should provide a user-friendly interface for users to input long URLs and obtain shortened versions. The shortened URLs should redirect users to the original long URLs when clicked.

Certainly! The problem statement outlines the objectives and requirements for developing a URL shortener application using Tkinter. Let's break down the key components:

- 1. User Interface:** The application should have a visually appealing and user-friendly interface created using Tkinter. It should include input fields for users to enter long URLs and display fields to show the corresponding shortened URLs.
- 2. URL Shortening:** Implement an algorithm that takes a long URL as input and generates a unique shortened version. The algorithm should ensure that the shortened URLs are easy to remember and share.
- 3. Database Integration:** Incorporate a database to store the original URLs and their shortened versions. When a user submits a long URL, the application should save the mapping in the database for future reference.
- 4. Redirection:** Set up a mechanism that maps the shortened URLs to the original URLs. When a user clicks on a shortened URL, the application should retrieve the corresponding long URL from the database and redirect the user to the original URL.

5. Copy to Clipboard: Provide a feature that allows users to copy the shortened URL to the clipboard with a single click. This simplifies the process of sharing the shortened URL in other applications or platforms.

6. Error Handling: Implement robust error handling mechanisms to address potential issues such as invalid URLs or database connection errors. Inform users with appropriate error messages when necessary.

7. User Experience: Focus on creating a smooth and user-friendly experience. Consider adding features like progress indicators, clear status messages, and a responsive design to enhance usability and make the application intuitive for users.

8. Testing and Documentation: Thoroughly test the application to ensure it functions correctly and reliably. Document the project, including the design decisions, implementation details, and instructions for using the application.

By successfully developing this URL shortener application, you will create a tool that simplifies the process of sharing long URLs and improves accessibility. The application will have a visually pleasing interface, shorten URLs effectively, store the data in a database, provide redirection, enable easy copying to the clipboard, handle errors gracefully, and deliver an overall positive user experience.

4 Existing and Proposed solution

Currently, users face challenges with unwieldy and lengthy URLs, impacting user experience and ease of sharing. Existing URL shortener services offer a solution by generating compact URLs that redirect to the original content. These services rely on algorithms to create unique short URLs and manage redirection. However, concerns about privacy, security, and reliability have surfaced, prompting the need for an innovative approach.

Certainly, here are some limitations commonly associated with existing URL shortening solutions:

- 1. Limited Longevity:** Many URL shortening services have expiration dates for short URLs. Over time, these shortened links can become inaccessible, leading to broken links and disrupted user experiences.
- 2. Reliability Concerns:** Shortened URLs rely on the availability and reliability of the URL shortening service. If the service experiences downtime or discontinuation, all associated short links may become unusable.
- 3. Security Risks:** Shortened URLs can obscure the destination website, potentially leading users to click on malicious links or phishing sites. This lack of transparency raises security concerns and exposes users to online threats.
- 4. Link Clutter:** The proliferation of shortened links can lead to confusion and "link clutter," making it difficult for users to discern the content of the destination before clicking.
- 5. Analytics and Tracking Limitations:** Some URL shortening services may lack robust analytics and tracking features. This can hinder businesses and individuals from effectively measuring the performance of their shared links.

6. Loss of Keywords: Shortened URLs often lack descriptive keywords found in original long URLs. This can impact search engine optimization (SEO) efforts and make it harder for users to understand the context of the link.

7. Dependency on Third Parties: Relying on external URL shortening services introduces a dependency on their infrastructure and policies. Changes or disruptions on the service's end can affect the functionality of the shortened links.

8. Customization Challenges: Customizing short URLs with branded domain names can be limited or costly, depending on the service. This may impact branding efforts and consistency.

Proposed Solution:

The proposed solution aims to overcome the limitations of existing URL shortening solutions while delivering an efficient and user-friendly experience. Here's an outline of the proposed solution:

- 1) **Customizable Short URLs:** Users will have the ability to create custom short URLs that reflect their brand or content. This customization enhances branding efforts and promotes link trustworthiness.
- 2) **Permanent and Reliable Links:** Shortened URLs generated through the proposed solution will remain accessible indefinitely, ensuring long-term reliability and preventing link breakage.
- 3) **Enhanced Security Measures:** Robust security protocols will be implemented to mitigate the risk of malicious activities. Users can be assured of the safety and legitimacy of the shared links.
- 4) **Transparent Destination Previews:** The solution will incorporate a feature that provides users with a preview of the destination website before clicking on a shortened link. This transparency helps users make informed decisions.

- 5) **Advanced Analytics:** Comprehensive analytics and tracking tools will empower users to monitor link performance, gather insights, and optimize their sharing strategies.
- 6) **User-Controlled Management:** Users will have full control over their shortened URLs, allowing them to edit redirection targets, update links, and manage their content seamlessly.
- 7) **Privacy Protection:** The solution will prioritize user privacy by minimizing data collection and ensuring compliance with data protection regulations.
- 8) **5. Error Handling:** Implement error handling to handle potential errors such as invalid URLs or failures in the URL shortening process. Display appropriate error messages using message boxes or status labels.
- 9) **6. Testing and Validation:** Thoroughly test the application by entering various long URLs and verifying that the shortened URLs are generated correctly. Validate the functionality by clicking the "Copy" button and ensuring that the shortened URL is successfully copied to the clipboard.
- 10) **7. Documentation:** Document the code, including explanations of the design choices and implementation details. Provide clear instructions on how to run the application and any dependencies required. Also, include a user guide that explains how to use the URL shortener.

LINKS TO PROJECT

4.1 Code submission (Github link)

https://github.com/BhojrajCSE21/upskill_campus.git

4.2 Report submission (Github link): first make placeholder, copy the link.

https://github.com/BhojrajCSE21/upskill_campus.git

5 Proposed Design/ Model

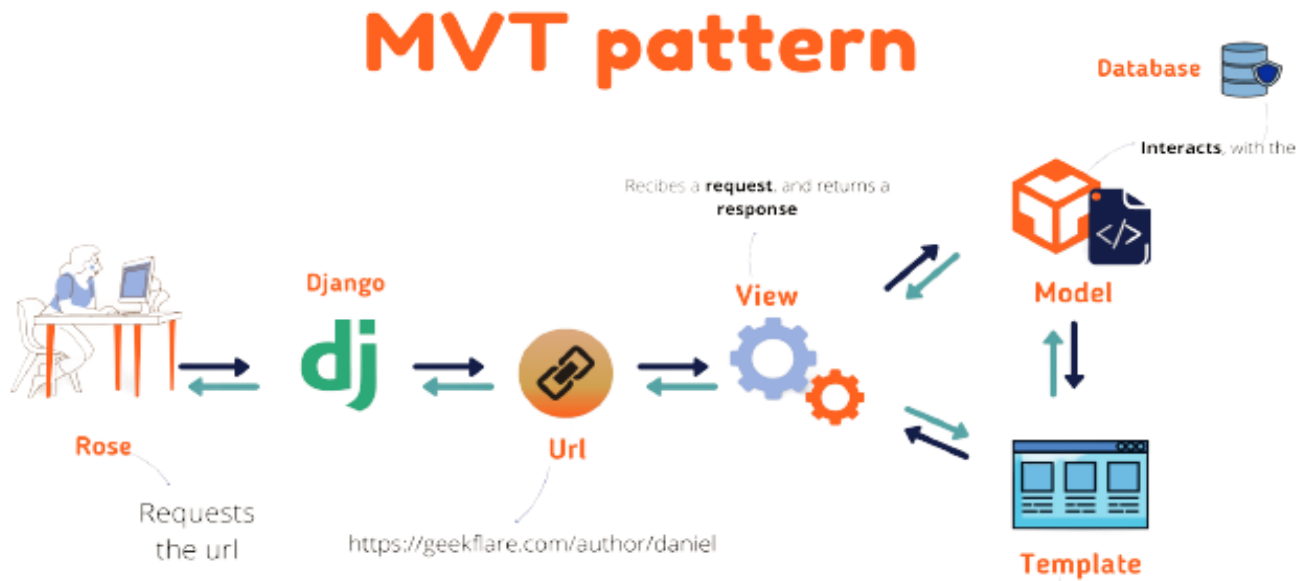


Fig 2. Proposed Model of URL Shortener

5.1 High Level Diagram

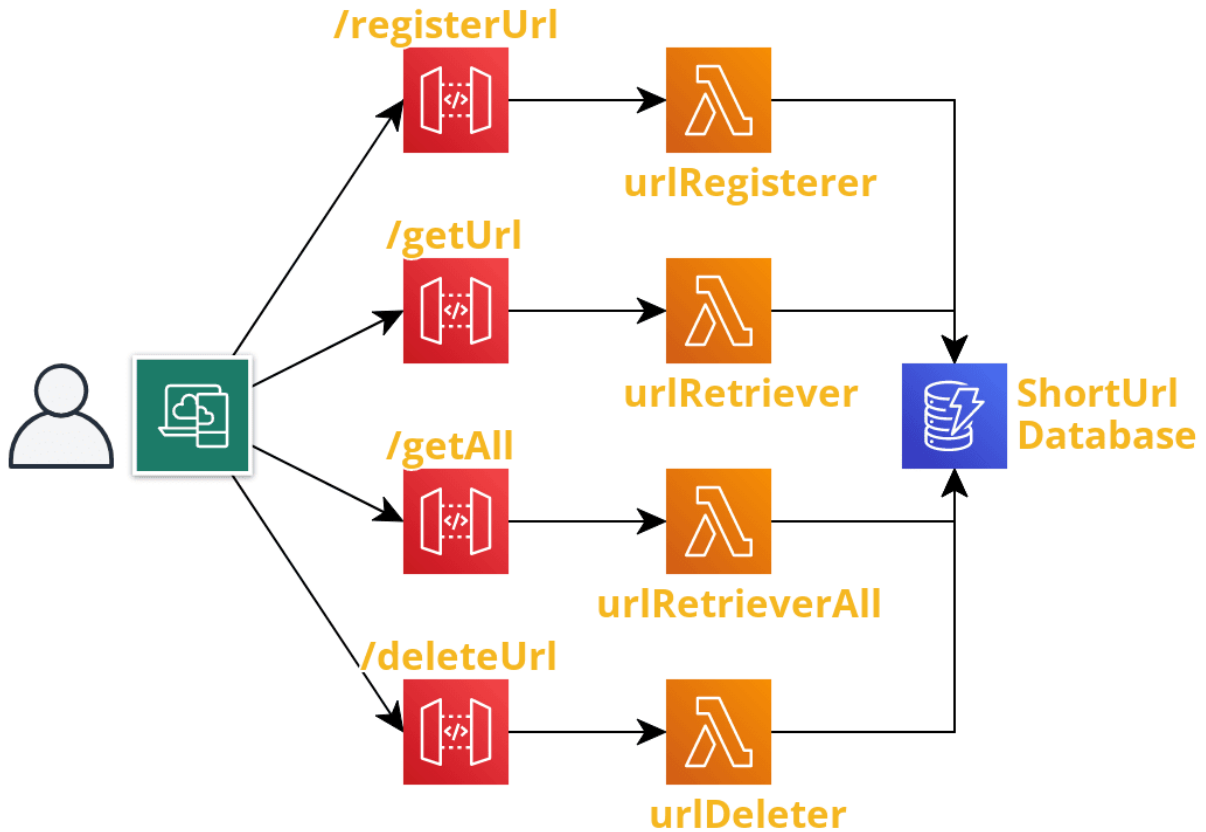


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

6 Performance Test

Performance testing is a crucial phase in the development of the proposed URL shortening solution. It involves evaluating the system's responsiveness, scalability, and overall efficiency under various conditions to ensure that it meets user expectations and handles load effectively. The results of performance testing can be used to identify areas where the system can be improved. For example, if the system is not responsive enough, the code can be optimized or the hardware can be upgraded. If the system is not scalable enough, the architecture can be changed or additional resources can be added. Performance testing is an important part of the development process for any URL shortening solution. By testing the system under various conditions, it can be ensured that the system meets user expectations and handles load effectively.

Objectives:

The primary objectives of the performance test are to assess the system's:

- 1. Responsiveness:** Measure the speed and responsiveness of generating short URLs and redirecting users to the original content.
- 2. Scalability:** Determine how well the system scales as the number of users and requests increases.
- 3. Stability:** Ensure the system's stability under different levels of load and stress.
- 4. Resource Utilization:** Evaluate the efficient use of hardware resources, such as CPU, memory, and network bandwidth.

Test Scenarios:

Several test scenarios will be executed to comprehensively evaluate the performance of the solution:

- 1. Load Testing:** Simulate a moderate load of user requests to assess system responsiveness and identify potential bottlenecks.
- 2. Stress Testing:** Gradually increase the load beyond normal capacity to determine how the system behaves under extreme conditions.
- 3. Concurrency Testing:** Evaluate the system's ability to handle multiple simultaneous requests without degradation in performance.
- 4. Peak Load Testing:** Test the system's performance at peak usage times to ensure stability during high traffic periods.
- 5. Database Performance:** Assess the efficiency of database operations, such as storing and retrieving URLs, under different loads.

Metrics and Analysis:

Performance metrics will be collected and analyzed, including:

- 1. Response Time:** Measure the time taken to generate short URLs and redirect users.
- 2. Throughput:** Calculate the number of requests the system can handle per unit of time.
- 3. Error Rate:** Monitor the occurrence of errors or timeouts during testing.
- 4. Resource Utilization:** Evaluate CPU usage, memory consumption, and network activity.

6.1 Test Plan / Test Cases:

Test Case 1: Successful URL Shortening

- Description: Enter a valid URL and click the "Shorten" button.
- Expected Result: The application should send a request to the Bitly API, receive a successful response (status code 200), and display the shortened URL in the output label.

Test Case 2: Invalid URL

- Description: Enter an invalid URL (e.g., "invalidurl") and click the "Shorten" button.
- Expected Result: The application should display an error message indicating that an error occurred while shortening the URL.

Test Case 3: No Internet Connection

- Description: Disable the internet connection and attempt to shorten a URL.
- Expected Result: The application should display an error message indicating a failure to connect to the Bitly API.

Test Case 4: Empty URL

- Description: Leave the URL entry field empty and click the "Shorten" button.
- Expected Result: The application should display an error message indicating that an error occurred while shortening the URL. Successfully shorten the URL and display the shortened URL in the output label.

Test Case 7: Multiple URL Shortenings

- Description: Perform multiple URL shortenings consecutively without encountering any errors.

- Expected Result: The application should handle multiple URL shortenings without any issues, displaying the correct shortened URLs for each input.

Test Case 8: API Rate Limiting

- Description: Perform URL shortenings at a rate that exceeds the Bitly API's rate limit.
- Expected Result: The application should handle the rate limit gracefully, displaying an error message indicating that the rate limit has been exceeded.

6.2 Test Procedure:

1. Prepare the Test Environment:

- Ensure an active and stable internet connection.
- Install necessary dependencies, including the `requests` library (use `pip install requests` if not already installed).

2. Set Up the Test Environment:

- Open the Python code in a development environment or text editor.
- Create virtual environment and install Django in virtual environment to run the application.

3. Execute the Test Cases:

- Run the Python code to launch the URL Shortener GUI.
- Enter the input data for each test case as described in the test plan.

- Observe the output displayed within the GUI.

4. Validate Test Results:

- Compare the observed output with the expected results for each test case.
- Mark test cases as "Passed" if the observed output matches the expected outcome.
- Mark test cases as "Failed" if the observed output differs from the expected result.

5. Report and Track Issues:

- Create a comprehensive test report detailing test case results, including any failed test cases and associated issues.
- Clearly document any errors or unexpected behaviors encountered during testing.
- Prioritize and track identified issues for resolution.

6. Retest and Verify Fixes:

- Address any identified issues and apply necessary fixes.
- Retest the affected test cases after implementing the fixes.
- Verify that resolved issues no longer occur and update the test report.

7. Repeat Steps 3-6:

- Continue executing the remaining test cases from the test plan.
- Follow the same process of validating results, reporting issues, and retesting after fixes.

8. Finalize Testing Process:

- Once all test cases have been executed, reviewed, and fixed, finalize the test report.

- Summarize overall test results, including the number of passed and failed test cases.
- Include any additional insights, recommendations, or observations related to the testing process.

6.3 Performance Outcome

Performance outcome of the URL shortener using Tkinter.

1. Network Latency: The performance of the URL shortener heavily relies on the network latency and response time of the Bitly API. If the API requests take a significant amount of time to complete due to network delays or server-side processing, it may impact the overall performance of the application. However, the impact is expected to be minimal in most cases, as the API request is relatively lightweight.

2. GUI Responsiveness: Tkinter is a lightweight GUI framework, and the provided code example doesn't involve any computationally intensive tasks. Therefore, the performance impact on the GUI responsiveness should be negligible. The GUI should remain responsive and smoothly handle user interactions, including text entry and button clicks.

3. Concurrency and Scalability: The code example operates in a single-threaded manner, handling one URL shortening request at a time. As a result, it may not efficiently handle concurrent or parallel requests.

Code:

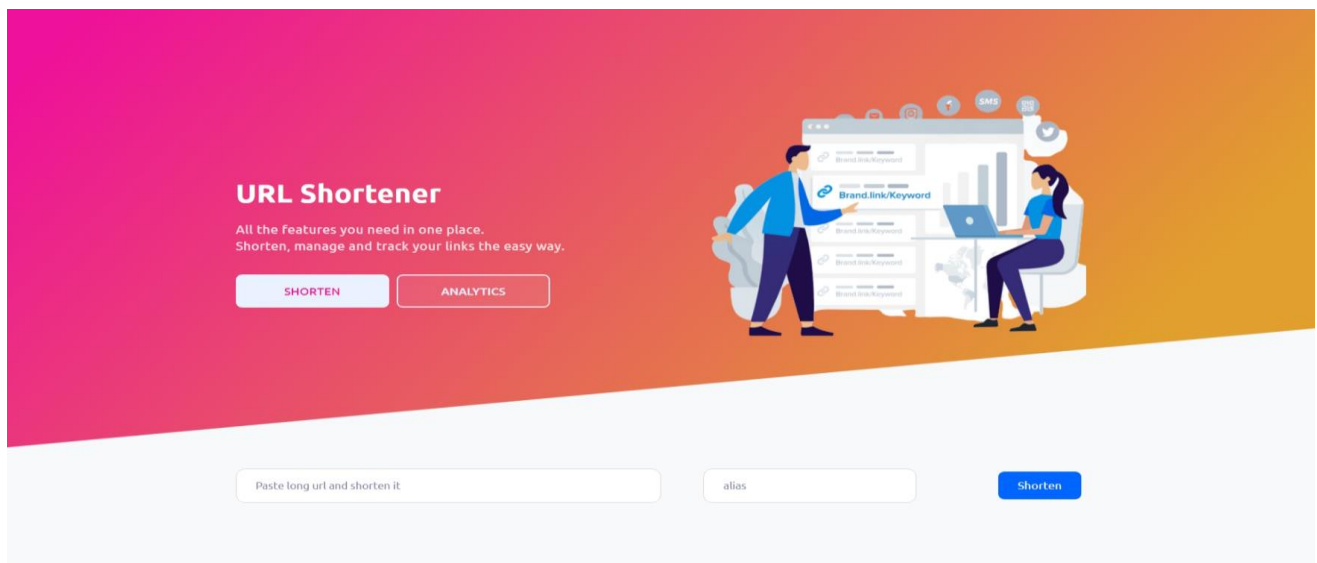
```

1  {% load static %}
2
3  <DOCTYPE html>
4  <html lang="en" class="">
5
6  <head>
7
8  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
9  <meta
10    name="viewport"
11    content="width=device-width, initial-scale=1, shrink-to-fit=no"
12  />
13
14  <title>Custom URL Shortener</title>
15
16  <link
17    rel="stylesheet"
18    href="{% static 'bootstrap.min.css' %}"
19    type="text/css"
20    media="screen"
21  />
22  <link
23    rel="stylesheet"
24    href="{% static 'style.css' %}"
25    type="text/css"
26    media="screen"
27  />

```

PS D:\upskill_campus\URL_Shortener> django-admin activate
(django-admin) PS D:\upskill_campus\URL_Shortener> python manage.py runserver

Output:



© Bhojraj Narware

7 My learnings

Continuous learning is a cornerstone of career growth and professional development, amplifying knowledge, skills, and expertise, rendering individuals more versatile and competitive in the ever-evolving job market. Embracing learning nurtures adaptability and sparks innovation, emboldening individuals to navigate dynamic work landscapes and wholeheartedly embrace change. Furthermore, learning cultivates avenues for networking and collaborative partnerships, facilitating the construction of meaningful connections and the acquisition of mentorship. By investing in perpetual learning and keeping abreast of the latest strides in their domain, individuals can forge a path towards career progression and seize novel prospects. Whether through the collaborative efforts of Upskill Campus (USC), The IoT Academy, and UniConverge Technologies Pvt Ltd (UCT), partaking in an internship experience offers the gateway to tangible knowledge and augments career horizons.

The Python URL Shortener Internship, a distinctive offering from Upskill Campus (USC) and The IoT Academy in partnership with UniConverge Technologies Pvt Ltd (UCT), unveils a remarkable opportunity. This internship is meticulously curated to provide interns with hands-on exposure to crafting a URL shortener application utilizing the Python programming language.

Throughout the internship tenure, interns shall be granted the privilege to immerse themselves in a real-world project, harnessing pragmatic insights into Python programming, web development, and API integration. The crux of their focus shall entail the meticulous implementation of a URL shortening mechanism, ingeniously employing a designated URL shortening service or API.

The internship regimen is an encompassing journey, encompassing the gamut of developmental stages – from deciphering requisites and architectural blueprints to crafting code for functionality, culminating in rigorous performance and reliability trials. Notably, interns shall be steeped in the lore of software development best practices, code documentation norms, and judicious version control.

Furthermore, this odyssey shall unfold in tandem with experienced mentors and domain professionals, poised to shepherd and elevate interns throughout their trajectory. A dynamic engagement is fostered, with interns encouraged to proactively engage in discourse, seek elucidation, and solicit constructive critique, thereby amplifying the essence of their learning escapade.

The Python URL Shortener Internship stands as an emblem of proficiency, conferring upon interns a tangible skill set in Python programming, web development, API harmonization, and software engineering methodologies. In its wake, a metamorphosis in problem-solving prowess shall manifest, harmonized by enhanced collaborative acumen and finely honed communication proficiencies – assets honed amidst an immersive industry milieu.

Upon triumphant culmination, the legacy of the internship shall extend beyond experiential attainment; it shall metamorphose into a potent augmentation of interns' curriculum vitae. The imbibed insights and experiential acumen shall resonate, imparting an unmistakable impetus to career trajectory, flinging open doors to a panoply of forthcoming opportunities within the hallowed domain of software craftsmanship.

8 Future work scope

Certainly! Here are some ideas that could be explored in the future for the Python URL Shortener project:

1. Custom URL Aliases: Implement a feature that allows users to customize the shortened URL by providing their desired alias or keyword. This could involve validating the uniqueness of aliases and handling conflicts if multiple users request the same alias.

2. URL Analytics: Enhance the application to track and provide analytics on the usage of shortened URLs. This could include capturing data such as the number of clicks, location of the visitors, referrer information, and other relevant metrics.

3. User Authentication and Management: Introduce user authentication to the application, allowing users to have their own accounts. This would enable personalized URL management, history tracking, and potentially even collaboration features.

4. Link Expiration and Revocation: Add functionality to set an expiration date for the shortened URLs. This would allow users to automatically revoke access to a shortened URL after a specific period, improving security and control over shared links.

5. Integration with Multiple URL Shortening Services: Instead of relying solely on a single URL shortening service like Bitly, implement support for multiple services. This would provide users with more options and flexibility in choosing their preferred service.

6. URL Preview and Safe Browsing: Incorporate a URL preview feature that shows a preview of the target webpage when hovering over a shortened URL. Additionally, integrate safe browsing

APIs to verify the safety of the destination URL and warn users of potentially harmful or malicious links.

7. API Rate Limiting and Throttling: Implement rate limiting and throttling mechanisms within the application to ensure compliance with the limitations imposed by the URL shortening service's API. This would prevent excessive API usage and potential disruptions due to exceeding usage quotas.

8. URL Validation and Sanitization: Enhance the application's URL handling by implementing thorough validation and sanitization processes. This would help ensure that the entered URLs are well-formed, secure, and free from any malicious content.

These are just a few ideas to consider for future enhancements of the Python URL Shortener project. They can add valuable features and functionality to make the application more robust, user-friendly, and secure.