Industrial Internship Report on

”Url Shortener”

Prepared by

Sakshi Kandur

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 Url Shortener Project provided by UCT. We had to finish the project including the report in 6 weeks’ time. My project was about The URL Shortener which allows users to input long URLs and generates shortened versions, which can be easily shared and accessed. I developed a URL Shortener using Python as the primary language. For the backend, I utilized Django, a Python framework known for its simplicity and scalability. On the frontend, I employed HTML, CSS, and JavaScript to create a user-friendly interface. This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface 3

2 Introduction 5

2.1 About UniConverge Technologies Pvt Ltd 5

2.2 About upskill Campus....10

2.3 Objective 12

2.4 Reference 12

2.5 Glossary....13

3 Problem Statement....14

4 Existing and Proposed solution 15

5 Proposed Design/ Model 17

5.1 High Level Diagram (if applicable) 18

5.2 Low Level Diagram (if applicable) 19

5.3 Interfaces (if applicable)....19

6 Performance Test 20

6.1 Test Plan/ Test Cases 21

6.2 Test Procedure....22

6.3 Performance Outcome....24

7 My learnings....25

8 Future work scope 26

1 Preface

Summary of the whole 6 weeks' work:

Week 1-2: Research and Planning

- Researched various technologies and frameworks suitable for building a URL Shortener.
- Planed the project scope, including desired features, user interface design, and database structure.

Week 3: Backend Development

- Implemented the backend logic using Django, focusing on URL routing, database models
- Tested backend functionality to ensure proper handling of URL shortening and redirection.

Week 4: Frontend Development

- Designed and developed the frontend interface using HTML, CSS, and JavaScript.
- Created forms for users to input long URLs and view their shortened versions.

Week 5: Integration and Testing

- Integrated the frontend with the backend, ensuring seamless communication between the two layers.
- Identified any bugs or issues to ensure the stability and reliability of the URL Shortener.

Week 6 : Deployment

- Deployed the URL Shortener project to a hosting platform, making it accessible to users.

This internship helps me in enhance my career development by providing invaluable practical experience, exposure to real-world challenges, and opportunities to apply theoretical knowledge in a professional setting.

Brief about My project:

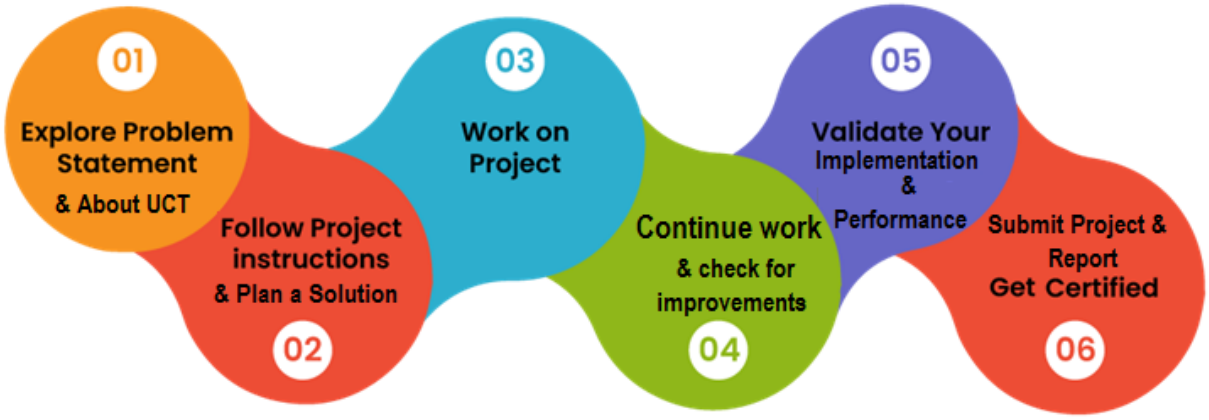
My project is a URL shortener, where the primary objective is to convert long URLs into short ones using Django for the backend and HTML, CSS, and JavaScript for the frontend to ensure user-friendliness. The process involves uploading a long or short website link, and upon clicking "generate URL," the system generates a shortened URL. Users can then copy and paste this URL into a new tab, which redirects them to the original website. Additionally, the backend includes administrative functionality, allowing admins to view the total number of URLs shortened and the corresponding original websites.

Opportunity given by USC/UCT.

Upskill Campus, powered by Uniconverge Technologies Pvt Ltd, offers a range of opportunities aimed at empowering individuals with the skills needed to succeed in today's competitive job market. Through their various programs and initiatives, Upskill Campus provides:

- Skill Development: They offer training programs and courses designed to enhance technical.
- Industry-Relevant Curriculum: Their curriculum is developed in collaboration with industry experts
- Internship Opportunities: They facilitate internships with leading companies, allowing learners to gain practical experience, build professional networks, and explore potential career paths.
- Mentorship and Support: Upskill Campus provides mentorship and support throughout the learning journey, helping learners overcome challenges, set goals, and achieve success.

How Program was planned



Throughout this project, I've gained invaluable insights and experiences that have contributed to my personal and professional growth. I've learned how to effectively utilize Python and Django for backend development, as well as HTML, CSS, and JavaScript for frontend design, enhancing my technical skills significantly.

I would like to extend my heartfelt gratitude to Nitin Tyagi Sir, Ankit Sir ,Jitesh Mathur Sir, and all uct members whose guidance and support were instrumental in navigating through the complexities of this project. Their expertise and encouragement have been invaluable, and I am deeply grateful for their mentorship.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

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

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

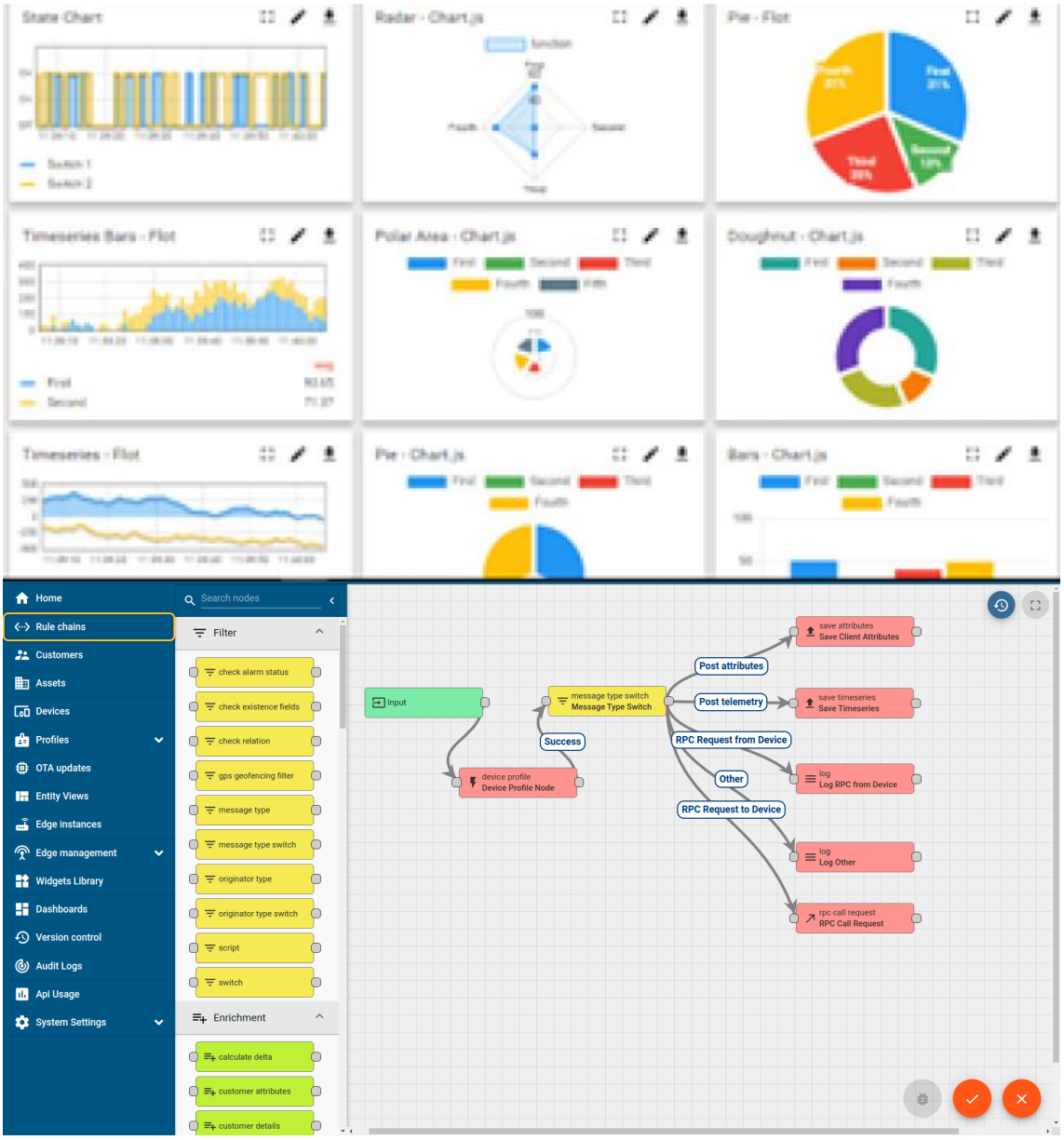
i. UCT IoT Platform(UCT Insight)

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

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

It has features to

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


Home
Rule chains
Customers
Assets
Devices
Profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
Search nodes
Filter
check alarm status
check existence fields
check relation
gps geofencing filter
message type
message type switch
originator type
originator type switch
script
switch
Enrichment
calculate delta
customer attributes
customer details
Input
message type switch
Message Type Switch
Success
device profile
Device Profile Node
Post attributes
Post telemetry
RPC Request from Device
Other
RPC Request to Device
save attributes
Save Client Attributes
save timeseries
Save Timeseries
log
Log RPC from Device
log
Log Other
rpc call request
RPC Call Request

ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

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

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

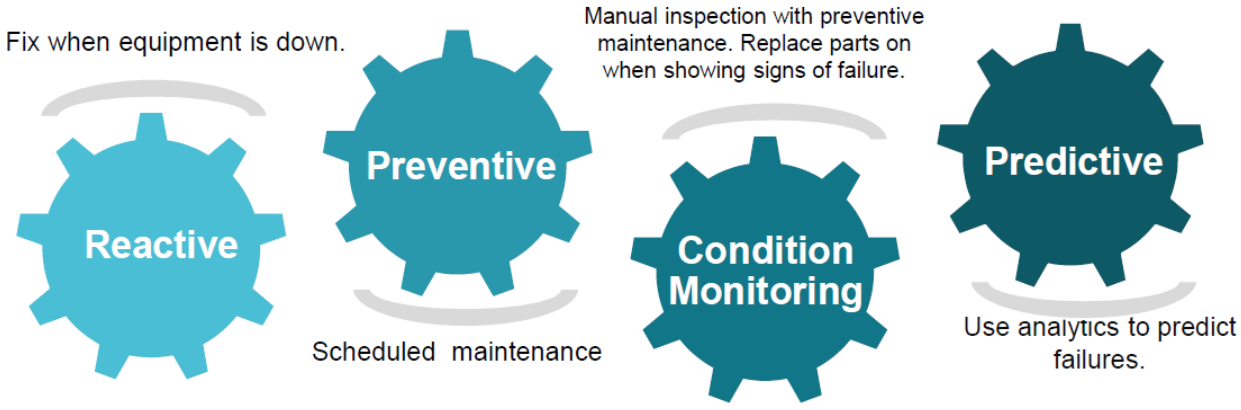
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

iii. LoRaWAN based Solution

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

iv. Predictive Maintenance

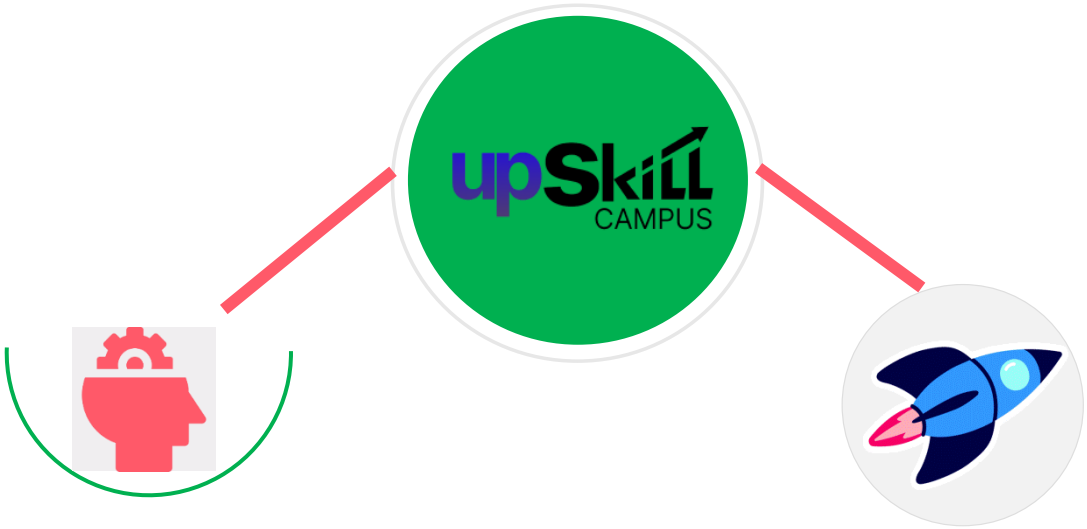
UCT isproviding Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

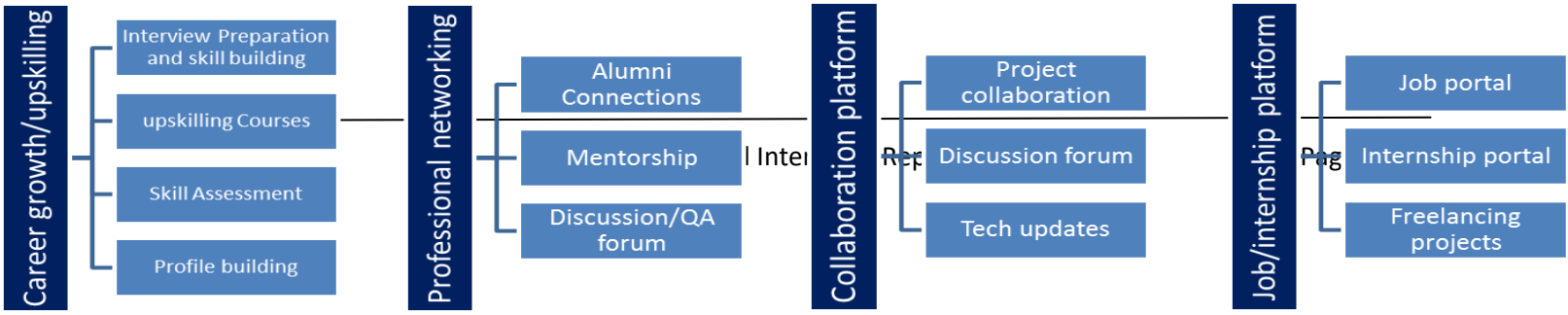
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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] chatgpt , Blackbox error solving guide

[2]

[3]

2.6 Glossary

Terms	Acronym
Redirection	-
Analytics	-
User Authentication	-
Backend	-
Database	-

3 Problem Statement

URL SHORTENER:

- In a digitally connected world, the length of URLs can often become unwieldy and inconvenient, especially when sharing them through various platforms such as social media, email, or messaging apps. This presents a significant challenge for users seeking concise and manageable ways to share links effectively.
- The problem lies in the need for a solution that can efficiently convert long URLs into shorter, more user-friendly versions, while ensuring reliability, security, and scalability.
- Therefore, the objective is to develop a URL shortener application using Django, a high-level Python web framework, which addresses these challenges by providing users with the ability to generate short URLs for long links.
- This offers a seamless user experience, allowing users to easily input long URLs and receive corresponding shortened versions.
- Furthermore, the application should include administrative functionality for monitoring and managing the shortened URLs.

Existing and Proposed solution

Provide summary of existing solutions provided by others, what are their limitations?

Hosting and Maintenance:

- Hosting the application on a personal server or cloud platform requires ongoing maintenance, including server updates, security patches, and database management.

Security Concerns:

- Despite Django's built-in security features, the application may still be susceptible to security vulnerabilities if not properly configured or maintained.

Limited Customization for Frontend:

- While HTML, CSS, and JavaScript offer flexibility for frontend development, the level of customization may be limited compared to more advanced frontend frameworks.

What is your proposed solution?

The proposed solution for the URL shortener project involves developing a web application using Django, a high-level Python web framework, along with HTML, CSS, and JavaScript for the frontend. Here's an overview of the proposed solution:

1. Backend Development with Django:

- Utilize Django to create the backend logic for the URL shortener application.
- Implement URL routing to handle incoming requests and map them to the appropriate views.
- Develop models to store information about the original long URLs and their corresponding shortened versions in a database.

2. Frontend Development with HTML, CSS, and JavaScript:

- Design and develop a user-friendly interface using HTML for structuring the web pages.
- Use CSS for styling the interface
- Incorporate JavaScript for client-side validation, dynamic page updates, and enhancing interactivity.
- Create forms for users to input long URLs and display the corresponding shortened versions.

3. Features and Functionality:

- Allow users to input long URLs and generate corresponding shortened URLs with a paste that url on new tab it will redirect to original website of url.

4.Deployment

- Implement scalability measures to handle a large volume of URL shortening requests and user traffic effectively.

What value addition are you planning?

Customizable short Url:

- Offering users the ability to customize their shortened URLs with meaningful aliases or keywords, enhancing branding and usability.

Browser Extension:

- Developing a browser extension or plugin that allows users to shorten URLs directly from their browser interface, streamlining the process and improving user convenience.

3.1 Code submission (Github link)

https://github.com/sakshikandur/upskillcampus/UrlShortener

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

4 Proposed Design/ Model

Given more details about design flow of your solution. This is applicable for all domains. DS/ML Students can cover it after they have their algorithm implementation. There is always a start, intermediate stages and then final outcome.

Here's a proposed model for the URL shortener project:

1. Backend Design:

In Models:

1. **URL:** This model stores information about the original long URLs and their corresponding shortened versions. It includes fields such as:
   - Original URL
   - Shortened URL
   - Click count

2.Views:

- Shorten URL View: Handles the logic for shortening long URLs and generating unique shortened versions.
- Redirect View: Redirects users from shortened URLs to their original destinations and updates the click count.

3.URL Routing: Define URL patterns for different views, such as "<str:pk>/".

2. Frontend Design:

HTML Templates:

- Shorten URL Form: Allows users to input long URLs and submit them for shortening.
- Shortened URL Display: Displays the shortened URL generated by the system.

CSS Stylesheets:

Apply styling to HTML elements for better user experience.

JavaScript:

- Form Validation: Validate input fields before submitting the form for URL shortening.
- Copy-to-Clipboard Functionality: Implement functionality to copy the shortened URL to the clipboard.

3. **Database Design:**
   I used a relational database management system i.e. dbsqlite to store URL data.

4.1 High Level Diagram (if applicable)

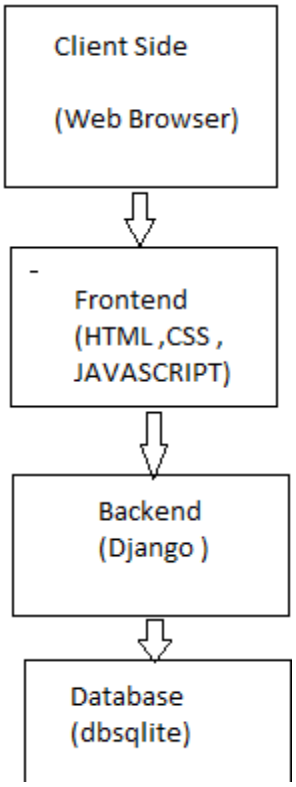
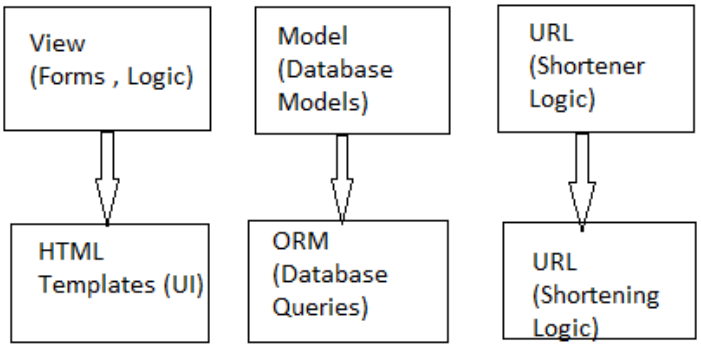


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

4.2 Low Level Diagram (if applicable)

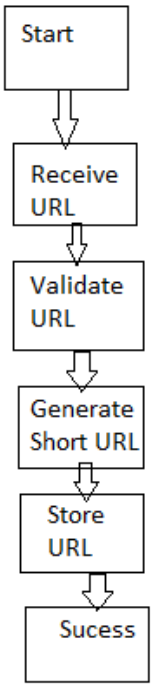


4.3 Interfaces (if applicable)

The flowchart illustrates the step-by-step process of URL shortening within the backend component of the system

1. detailing actions such as receiving a URL,
2. validating it
3. generating a shortened URL
4. storing it in the database
5. providing feedback to the user upon successful completion.

Each step is represented by a specific process, with arrows indicating the flow of control from one step to the next, ultimately leading to either successful completion or error handling.



5 Performance Test

In performance testing, the constraints are the limitations or boundaries within which the system must operate efficiently.

These constraints could include factors such as response time, throughput, concurrency, and scalability.

Identifying these constraints is crucial as they determine the system's ability to handle real-world usage scenarios and meet user expectations in terms of speed, reliability, and capacity.

In the design of a URL shortener system, several strategies can be employed to address performance constraints:

1. Response Time: The system should aim to generate shortened URLs and redirect users to the original URLs swiftly.
2. Scalabilty : Designing the system to be horizontally scalable allows it to accommodate increasing loads by adding more resources or scaling out across multiple servers.

3. Throughput: To ensure high throughput, the system should be capable of handling a large number of URL shortening requests concurrently.

5.1 Test Plan/ Test Cases:

Test Cases for URL Shortener:

1. **URL Shortening Functionality**:

Description: Verify that the system accurately shortens long URLs submitted by users.

Preconditions: User is authenticated and navigates to the URL shortening page.

Steps:

1. Enter a long URL into the input field.

2. Click the "Shorten" button.

Expected Results: Shortened URL is generated and displayed to the user.

4. **URL Redirection:**

Description: Verify that shortened URLs redirect users to their original destinations.

Preconditions: Shortened URL is valid and active.

Steps:

1. Enter the shortened URL into the browser's address bar.

2. Press Enter or click Go.

Expected Results: User is redirected to the original long URL.

5. **Error Handling** :

Description: Verify that appropriate error messages are displayed for invalid inputs or errors.

Preconditions: User submits invalid input or encounters an error condition.

Steps:

1. Enter an invalid URL format into the input field.

2. Click the "Shorten" button.

Expected Results: Error message is displayed indicating the invalid input.

6. **Reliability:**

Description: Assess system reliability under stress conditions.

Preconditions: System is subjected to stress testing scenarios.

Steps:

1. Increase load beyond normal operating conditions.

2. Monitor system behavior for errors, crashes, or downtime.

Expected Results: System remains stable and responsive, with no critical failures observed.

5.2 Test Procedure:

Test procedures for a URL shortener involve the step-by-step instructions for executing each test case to validate the system's functionality, performance, and reliability.

1. **Database Integrity**:

Steps:

1. Shorten multiple URLs and verify that they are stored correctly in the database.
2. Delete or modify records in the database and observe the impact on the system's behavior.

Expected Result: URL data is accurately stored, retrieved, and manipulated in the database without data corruption.

2.Cross-Browser Compatibility:

Steps:

1.Access the URL shortener system using different web browsers (e.g., Chrome, Firefox, Safari, Edge).

2.Perform URL shortening and redirection tests on each browser.

Expected Result: System functions consistently and reliably across various web browsers without compatibility issues.

3. URL Redirection:

Steps:

1. Copy a shortened URL from the system.

2. Paste the shortened URL into the browser's address bar.

3. Press Enter or click Go.

Expected Result: User is redirected to the original long URL.

4.User Authentication (if applicable):

Steps:

1.Attempt to access the URL shortening functionality without authentication.

2.Enter valid credentials and authenticate as a registered user.

3.Repeat URL shortening and redirection tests.

Expected Result: Unauthenticated users are prompted to log in, and authenticated users can successfully use the URL Shortener.

5.3 Performance Outcome

1. Users can fastly generate shortened URLs and access original links without delay
2. The system can handle many users at once, ensuring it doesn't slow down or crash system.
3. As more users join, the system can expand smoothly and can give the user what they want .
4. It remains consistent even with lots of people using it simultaneously, guaranteeing reliability.
5. User can handle the short url easily especially when they are sharing it will be easy to share short url.
6. By focusing on these performance aspects, a URL shortener system can deliver a fast, efficient, and can enhance user satisfaction

6 My learnings

Through this overall URL shortener project, I have gained valuable experience in full-stack web development, honing skills in backend development with Django, frontend design with HTML/CSS/JavaScript, and database management.

This hands-on experience equips me with practical knowledge of building scalable, user-friendly web applications. Opportunities stemming from this project include potential roles in software engineering, web development, or backend/frontend development positions in industries ranging from technology startups to established companies seeking innovative solutions for web-based services.

I also obtained a pathway to develop and optimize real-world applications like a URL Shortener, enhancing my practical knowledge throughout this project. I am sincerely thankful to Upskill Campus and Uniconverge Technologies Pvt. Ltd. for providing me with the opportunity to develop myself and preparing me to confront future challenges I may encounter.

7 Future work scope

1. Implementing analytics features to track usage statistics For example , the number of clicks on shortened URLs, geographic location of users, and popular URLs. This data can provide valuable insights for understanding user behavior.

2. Developing dedicated mobile applications for iOS and Android platforms, providing users with convenient access to the URL shortening service on their Smartphones and tablets.