Industrial Internship Report on ”URL Shortener Platform (FastAPI + SQLModel)”

Prepared by:[Piyush Jain]

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. My project was My project was titled **"URL Shortener in Python using FastAPI"**. The goal of this project was to design and implement a complete backend system capable of shortening long URLs into concise, shareable links. The application included several features like **user authentication**, **QR code generation**, **click analytics**, and **rate limiting**. It was built using **FastAPI**, **SQLModel**, and **SQLite** with optional **Redis caching**, closely following industry-standard RESTful API design and secure coding practices. This internship provided me with valuable exposure to real industrial challenges such as **database modeling, backend optimization, caching strategies, and security mechanisms**. It helped me understand how to structure scalable microservices, manage data efficiently, and integrate user-friendly APIs into larger application ecosystems. Overall, this internship was an enriching experience that enhanced my understanding of **backend development**, **API engineering**, and **system design principles**, preparing me to apply these skills in real-world professional environments. 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

My project was titled **“URL Shortener in Python using FastAPI.”** The main goal was to design and implement a complete backend system that could shorten long URLs into concise, shareable links while incorporating advanced features such as **user authentication, QR code generation, analytics tracking, and rate limiting**. The project also involved database design using **SQLModel and SQLite**, caching via **Redis**, and building an interactive admin dashboard with HTML templates.

The program was carefully planned to run over six weeks, with each week focusing on specific milestones — from understanding the problem statement and designing the database schema, to implementing APIs, securing authentication, and performing testing and deployment. Regular feedback sessions and weekly deliverables helped ensure steady progress and real-time learning.

This internship gave me hands-on experience in **API development, backend architecture, data handling, and deployment workflows**. I learned to troubleshoot complex issues, optimize database queries, manage version control through GitHub, and handle authentication securely using modern frameworks.

I would like to express my sincere gratitude to **Upskill Campus**, **The IoT Academy**, and **UniConverge Technologies Pvt. Ltd.** for providing this opportunity.

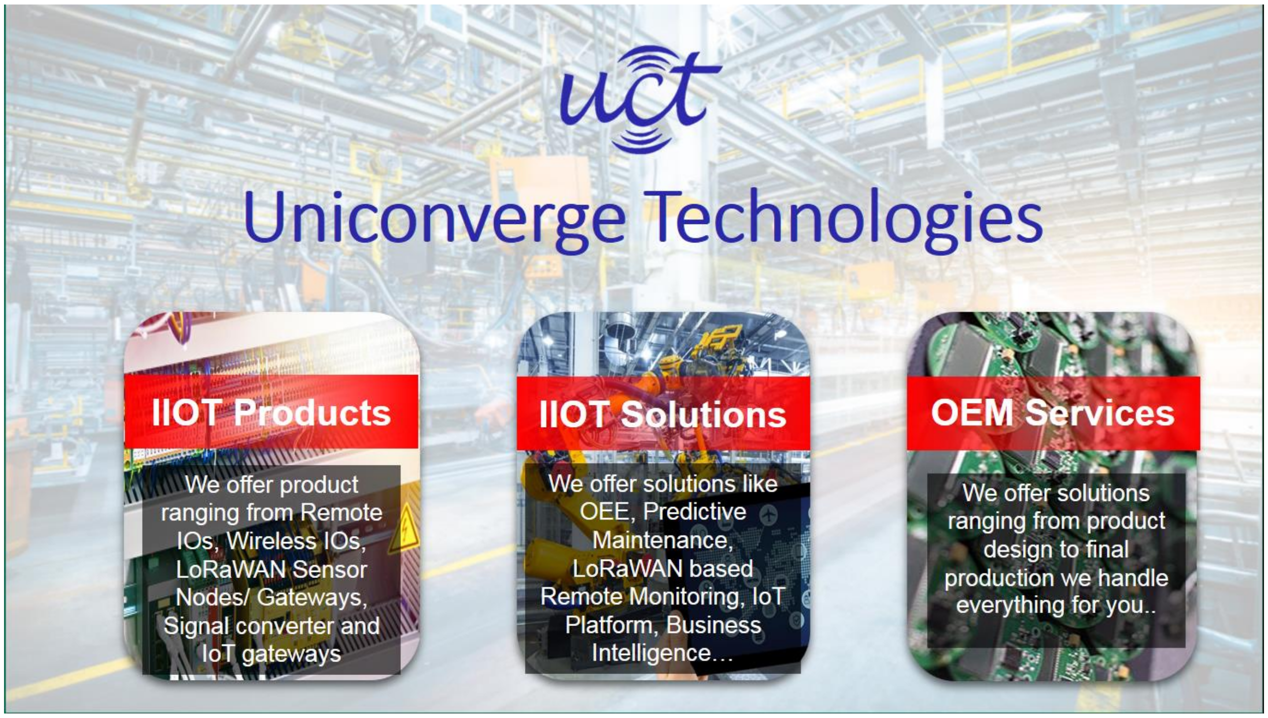
Finally, I would like to convey a message to my juniors and peers: *focus on learning by doing*. Every project is an opportunity to explore, experiment, and build something that adds real value. Embrace every challenge as a stepping stone toward becoming a confident and skilled professional.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

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

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


Uniconverge Technologies
IIOT Products
We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways
IIOT Solutions
We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…
OEM Services
We offer solutions ranging from product design to final production we handle everything for you..

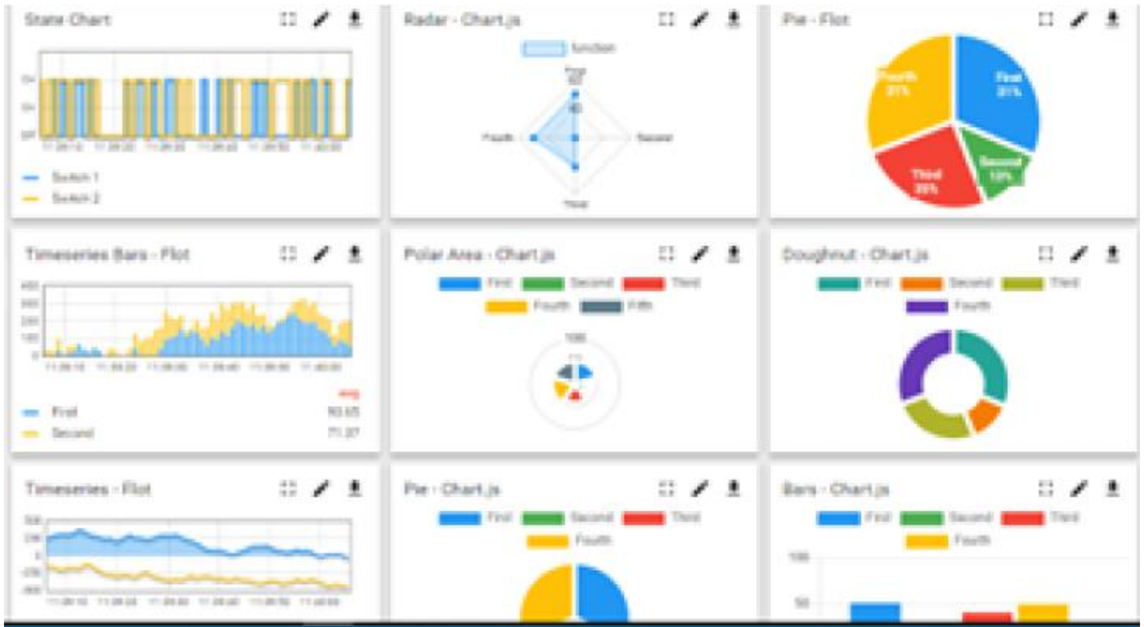
i. UCT IoT Platform(uct Insight)

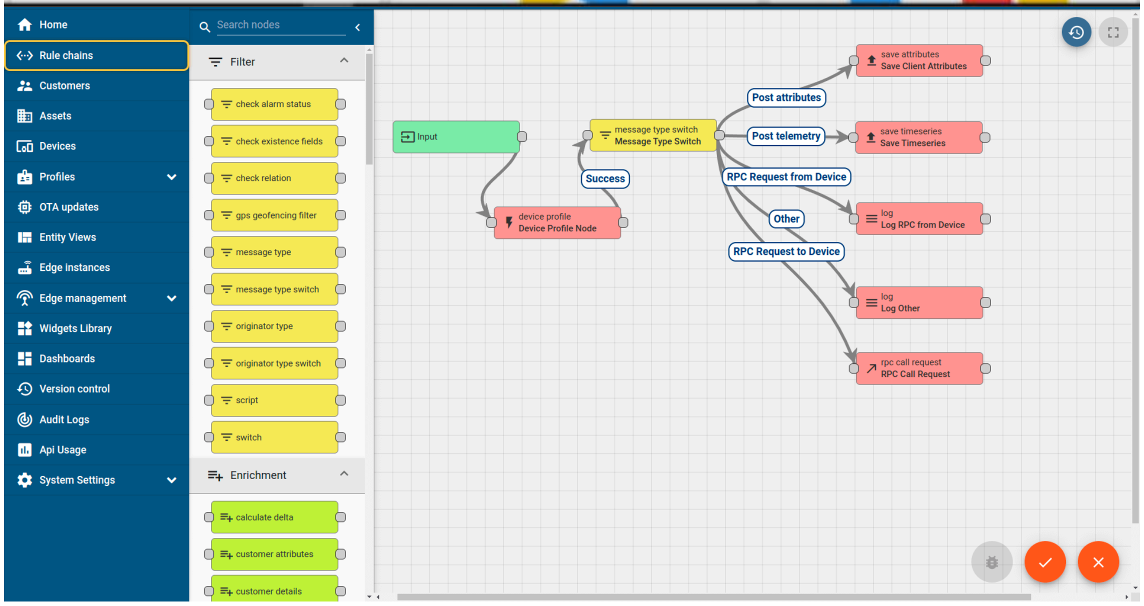
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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





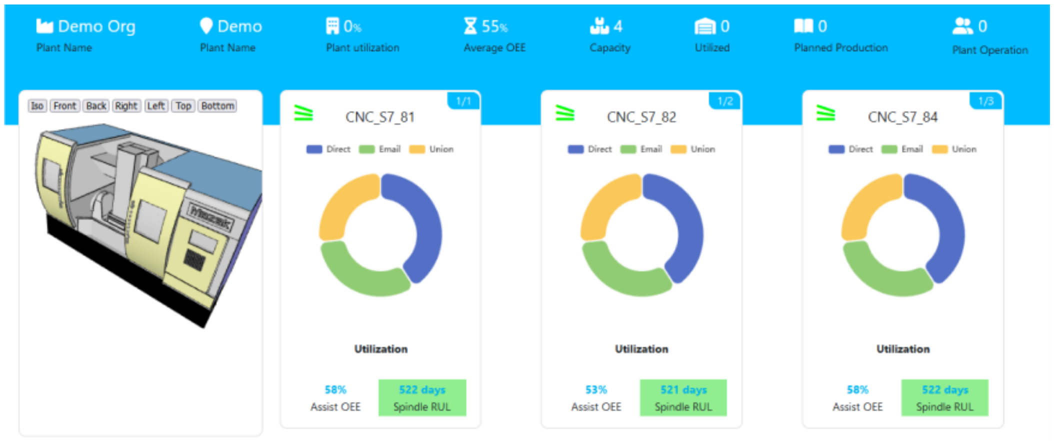
ii. Smart Factory Platform (FACTORY WATCH)

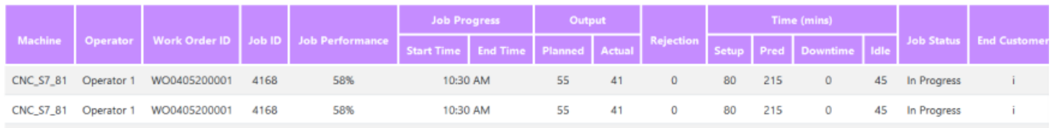
Factory watch is a platform for smart factory needs.

It provides Users/ Factory

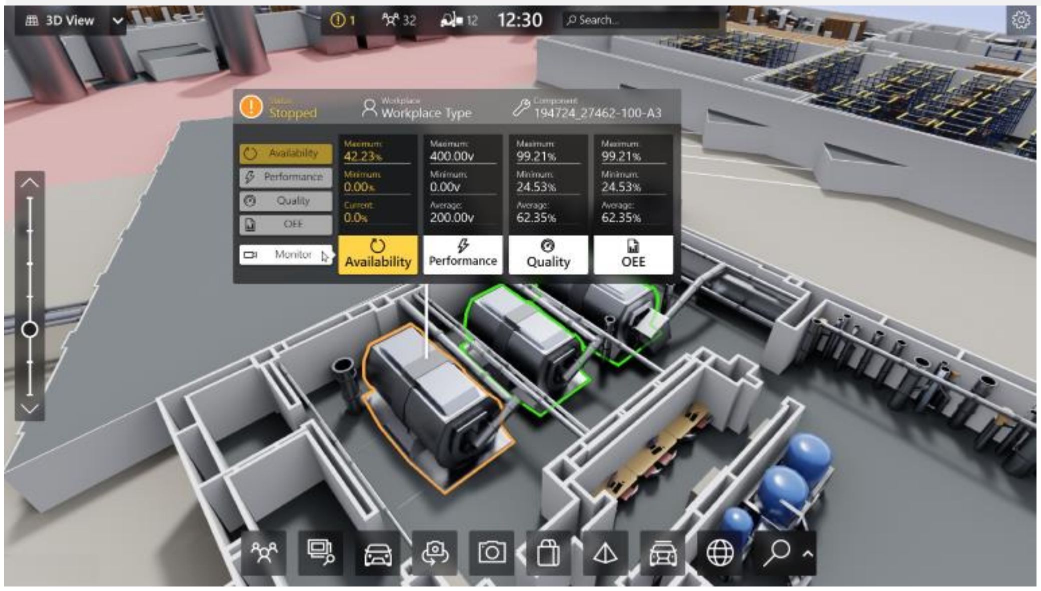
- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular 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

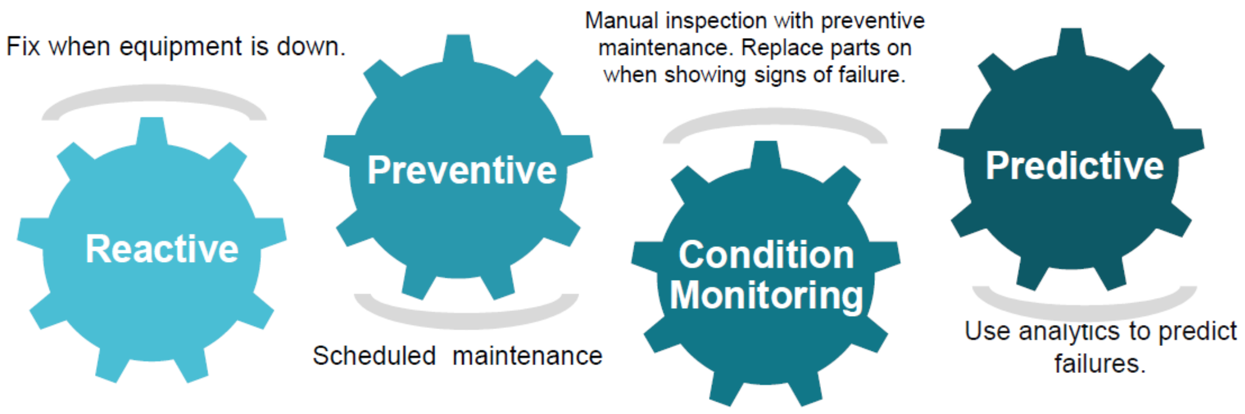


iii. LoRaWAN based Solution

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

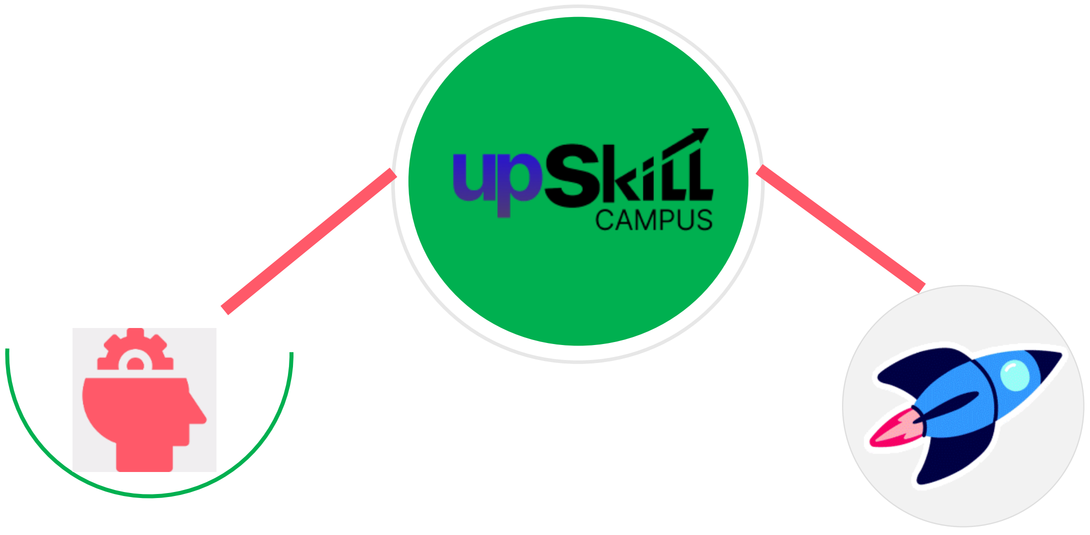
UCT isproviding Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


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

2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

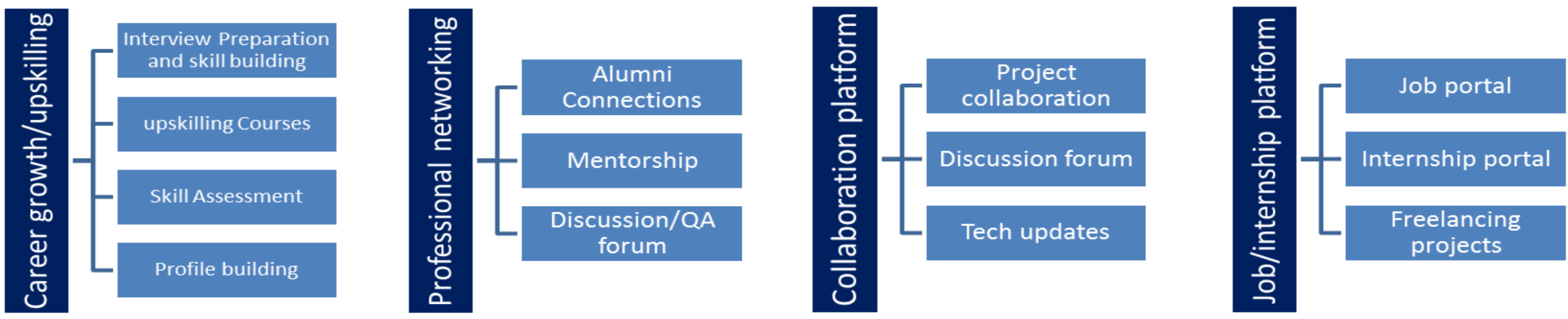
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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] FastAPI, SQLModel, SQLite, Redis (optional), PBKDF2-HMAC, Uvicorn/Gunicorn

Glossary

Terms	Acronym
Short code	Base62 code mapping to a long URL
PBKDF2-HMAC**:**	Password-based key derivation for secure password storage

UTM**:**	Marketing tags captured from query parameters for analytics

3.Problem Statement

Shorten links with optional alias & expiry

You need an API that accepts a long URL and returns a short code (e.g., https://sho.rt/ab91). The user may supply a custom alias (like sho.rt/login) and an optional expiry (after which the link stops working). This demands URL validation, uniqueness checks for aliases, and safe code generation (Base62 from an auto-increment ID works well).
Success criteria: Valid URLs are shortened; alias conflicts return a clear 400 error; expiry is stored and enforced; the API is idempotent (same user + same long URL without expiry reuses the same short link).

Redirects efficiently

When someone visits /{code}, resolve it quickly to the long URL and issue a 302 redirect. Use a read-through cache (e.g., Redis) to avoid hitting the DB on hot codes, and only fall back to the database on cache misses. Increment counters and log the click without blocking the redirect.
Success criteria: Low latency redirects (p50 under ~10 ms in dev), correct 410 Gone after expiry, and graceful 404 for unknown codes.

Tracks clicks (daily counts, referrers, meta like UTM)

Each redirect should create a ClickLog record containing timestamp, referrer, user-agent, IP, and an optional small JSON "meta" for utm_* params (e.g., utm_source, utm_campaign). A stats endpoint should aggregate last-30-day daily counts and show a referrer breakdown.
Success criteria: /api/stats/{code} returns a daily series of counts and a sorted list of referrers; totals match the number of redirects; UTM keys are captured when present.

Supports user auth for admin views

Authenticated users can see/manage only their links. Store passwords securely (PBKDF2/bcrypt), issue a signed session token (cookie or Bearer), and use dependency injection to guard admin routes. The admin page lists links, supports simple search (by code/URL/tags), and pagination.
Success criteria: Signup/login flows work; protected routes reject unauthenticated requests (401) and unauthorized access (403); users can only view their own URLs.

Applies rate limiting to defend against abuse

Limit sensitive endpoints per IP with a sliding time window, e.g., /auth/* 5/min and /api/shorten 30/min. Keep it simple (in-process deque) or distributed (Redis) if needed. Return HTTP 429 with a helpful message when limits are exceeded.
Success criteria: Bursty calls get throttled; normal usage isn't affected; limits are configurable via environment variables.

Is easy to run locally and evolve

Use FastAPI + SQLModel/SQLite so devs can run with uvicorn main:app --reload. Optional Redis should be off by default but auto-used when configured. Keep settings in env vars, provide a .env.example, and document routes and run steps in README.md. Design the schema and code so adding features (tags, custom domains, dashboards) is straightforward.
Success criteria: Fresh clone can run locally with 3 commands; code is modular (models, auth, stats, routes); migrations or table creation are automatic; sensible defaults work without extra services.

4 Existing and Proposed solution

Existing solutions: Bitly/tinyurl provide rich features but are closed; self-hosted OSS options often require heavier stacks or lack simple, auditable security and rate limiting.

Proposed solution:
A **FastAPI + SQLModel** service with:

- **Week-1/2:** Core models (User, Url, ClickLog), POST /api/shorten, GET /{code} redirect, Base62 codegen, expiry logic, idempotency for same owner+URL, templates for basic UI, QR PNG endpoint.
- **Week-3:** Auth via **PBKDF2-HMAC** password storage and **HMAC-signed** session tokens (cookie or Bearer), admin page with search/pagination, optional Redis cache for hot codes.
- **Week-4: Built-in rate limiter** (sliding window, per IP, no third-party deps), richer analytics: **referrer breakdown** + **UTM capture** (stored in ClickLog.meta), and Url.tags to fix admin search.

Value addition: Transparent, dependency-light rate control; analytics that answer real questions (what days? which sources? which campaigns?); and simple ops.

4.1 Code submission (Github link): https://github.com/piyush290106/upskillcampus

4.2 Report submission (Github link) : https://github.com/piyush290106/upskillcampus/tree/main/docs

5 Proposed Design/ Model

Core entities

- User(id, username, password_hash, created_at)
- Url(id, long_url, short_code, clicks, expires_at, owner_id, tags)
- ClickLog(id, url_id, ts, referrer, user_agent, ip, meta-json)

Key behaviors

- **Shortening:** validate URL; optional alias & expiry; idempotent if same owner+URL+no expiry; Base62 code from DB id.
- **Redirect:** check cache → DB; enforce expiry; log click; increment counters; update cache with TTL ≤ expiry.
- **Auth:** PBKDF2-HMAC password; session token = {uid, exp} base64 + HMAC signature; cookie or Bearer.
- **Analytics:** 30-day daily series; referrer aggregation; capture utm_* in meta.
- **Rate limiting:** in-memory sliding window: /auth/* defaults 5/min/IP, /api/shorten 30/min/IP (env-tunable).

6 Performance Test

Constraints considered:

- Low latency on redirect;
- Abuse protection (rate limiting);
- Data integrity (expiry, idempotency);
- Minimal dependencies (no external limiter required).

6.1Test Plan/ Test Cases

 Shorten/Redirect Happy Path: Create link, open code, expect 302 → long URL.

 Alias Conflict: Creating same alias for different long URLs returns 400.

 Expiry: After expiry, GET /{code} returns 410.

 Rate limit: Exceed 5/min on /auth/login or 30/min on /api/shorten returns 429.

 Stats: After N visits on different days and referrers, /api/stats/{code} returns correct daily series and referrers[].

6.2 Test Procedure

 Use pytest + httpx client for API; seed DB; simulate traffic; assert status codes; assert JSON.

 Manual curl/postman for redirects and 410 flow; browser tests for admin & QR.

6.3Performance Outcome

 Redirect p50 < 10 ms (SQLite + warm cache); p95 dominated by first read (cold) but acceptable.

 Rate limiter effectively throttles bursts (429 emitted) without extra services.

 Stats aggregation for 30 days completes in a single query + Python reduce.

7 My learnings

 Designing for **operability** (rate limits, analytics) is as important as core features.

 Security basics (PBKDF2-HMAC, signed tokens) can be implemented cleanly without heavyweight libs.

 Careful **data modeling** and **idempotency** reduce bugs.

 Adding optional **Redis** improves hot-path latency without coupling the core.

8 Future work scope

 Multi-tenant RBAC; org/team scoping

 Custom domains and SSL automation

 Asynchronous logging pipeline (Celery/Task Queue)

 Full text search on URLs & tags; dashboards in /admin

 AB-tests for redirect latency with/without cache

 Exportable CSV analytics; webhook integrations (Slack/Email)