VAJRA AND ODIN CRAWLER

A PROJECT REPORT

Submitted by,

Manur Yashas Sreevatsa - 20211CSE0275

Under the guidance of,

Dr. Anand Prakash

Associate Professor, School of Computer Science and Engineering

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

APRIL 2025

CERTIFICATE

This is to certify that the Internship/Project report **"VAJRA AND ODIN CRAWLER"** being submitted by "MANUR YASHAS SREEVATSA" bearing roll number "20211CSE0275" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr.ANAND PRAKASH
Associate Professor
PSCS / PSIS
Presidency University

Dr. MOHAMMED ASIF
HoD
PSCS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor - Engineering
Dean –PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled "**VAJRA AND ODIN CRAWLER"** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of my own investigations carried under the guidance of **Dr. Anand Prakash,** Associate Professor**, Prsidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

MANUR YASHAS SREEVATSA
20211CSE0275

INTERNSHIP COMPLETION CERTIFICATE

ASTRAEUS NEXT GEN PRIVATE LIMITED

CIN Number: U62099PN2024PTC233411

AVANEESH SNO 7/1 (P) 7/2 NR, KOKAN EXPRESS, KOTHRUD, PUNE- 411038

Ref No. Date:21/05/2025

Recommendation Letter for Manur Yashas Sreevatsa

To Whom It May Concern,

I am pleased to write this recommendation letter for Manur Yashas Sreevatsa, who interned at Astraeus Next Gen Pvt. Ltd. from January 16th, 2025, to May 20th, 2025. During his time with us, Manur worked on two of our key technology platforms, Odin and Vajra, and consistently displayed a high level of professionalism, technical skill, and initiative.

Manur's primary responsibilities involved frontend development and API integration for both projects. His contributions included:

1. Building and refining responsive user interfaces for the Odin platform, which functions as an AI-powered intelligence engine for company and geopolitical data aggregation.
2. Developing dynamic frontend components and implementing seamless API integration to ensure real-time data exchange and operational fluidity.
3. Working on Vajra's traffic optimization dashboard for IXPs, assisting with the creation of a user-friendly interface and integrating data pipelines for real-time network visualization.
4. Collaborating closely with the backend and AI/ML teams to ensure all visual and data elements were aligned, optimized, and scalable.

Manur demonstrated excellent problem-solving skills and adaptability, delivering quality outputs on tight timelines. His attention to detail, proactive communication, and ability to work independently and in teams made him a valuable asset to our engineering team.

I am confident that Manur Yashas Sreevatsa will be a strong contributor in any future endeavour he undertakes and will bring value to any technical team or organization.

Please feel free to contact me at anushdhavale@astraeusnextgen.com for any further information.

Sincerely,
Anush Dhavale Chairman &
Co-Founder
Astraeus Next Gen Pvt. Ltd.

ABSTRACT

This report presents the work undertaken during my internship at **Astraeus Next Gen Pvt. Ltd.**, where I contributed as a **Full Stack Web Development Intern**. The internship involved the design and development of three innovative projects—**Vajra and Odin Crawler**—each addressing real-world challenges in networking, web crawling, and software debugging.**Vajra** is a network optimization and cybersecurity platform that intelligently reroutes internet traffic across Internet Exchange Points (IXPs) to reduce congestion and improve performance. It features a dynamic pricing model based on actual improvements in latency and data speed.**Odin Crawler** is a smart, subscription-based web crawling system that allows users to search for content using keywords and receive a curated list of relevant links. It offers a 3-day trial and restricts access post-trial using PayPal integration, enabling controlled and monetized crawling.Together, these projects reflect the practical application of full stack development skills and AI integration to build scalable, intelligent systems. This report outlines the objectives, methodologies, and problem-solving approaches adopted throughout the internship.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presiency School of Computer Science and Engineering & Presiency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and Dr. "Mohammed Asif T",Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. ANAND PRAKASH,** Associate Professor, Presdiency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship/University Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Project Coordinator **Mr. Jerrin Joe Francis** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

MANUR YASHAS SREEVATSA

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

The rapidly evolving landscape of technology demands scalable, intelligent, and efficient solutions across diverse domains. During my internship at **Astraeus Next Gen Pvt. Ltd.**, I had the opportunity to contribute to three forward-thinking projects that reflect this need—**Vajra and Odin Crawler**. These projects were developed as part of my role as a **Full Stack Web Development Intern**, with the goal of addressing key challenges in networking, data extraction, and software debugging.

Vajra is a network optimization platform focused on enhancing data routing efficiency across Internet Exchange Points (IXPs). It intelligently identifies underutilized IXPs to reroute traffic, thereby improving latency and speed. The platform also integrates a performance-based billing model, offering enterprises a practical and measurable solution for traffic management.

Odin Crawler addresses the need for efficient and monetized web data extraction. It allows users to search for content via keywords and fetches up to 20 relevant URLs. Integrated with PayPal for subscription management, it ensures controlled access through a 3-day trial and subscription model.

This report provides a comprehensive overview of the problems tackled, the technical approaches used, and the value delivered through these three projects. It demonstrates the application of full stack development principles, backend engineering, and AI integration in solving modern-day software challenges.

CHAPTER 2
LITERATURE SURVEY

The Vajra – Network Optimization Platform was built on a robust theoretical foundation, combining classical graph algorithms with modern network routing protocols to ensure intelligent, scalable, and dynamic traffic optimization. At the core of its traffic rerouting engine lies Dijkstra's Algorithm [1], a time-tested method used to compute the shortest path between nodes in a graph. This is particularly useful in Vajra for identifying optimal routes between Internet Exchange Points (IXPs) under static conditions. However, in real-world scenarios where network latency and congestion levels are in constant flux, Dijkstra's static assumptions are insufficient. To address this, the platform integrates the Bellman-Ford Algorithm [2], which recalculates paths even when link costs change dynamically—ideal for adapting to fluctuating IXP performance in near real-time. For long-term scalability, the Floyd-Warshall Algorithm [3] is being explored as a dynamic programming approach to calculate shortest paths between all node pairs, enabling support for a fully meshed IXP environment. The platform's foundational routing principles also borrow heavily from the Border Gateway Protocol version 4 (BGP-4) [4], which remains the backbone of interdomain internet routing. Vajra layers intelligent, real-time rerouting decisions atop BGP's static routing base, improving latency and throughput without disrupting global routing consistency. Design decisions within Vajra were influenced by concepts from Computer Networking: A Top-Down Approach [5], guiding the distinction between control and data planes. The Spanning Tree Protocol (STP) [6] was incorporated to prevent routing loops in clustered IXPs, ensuring both fault tolerance and broadcast suppression. Additional reference to Routing in the Internet [7] informed trade-offs between link-state and distance-vector protocols. For large-scale dynamic environments, integration with Open Shortest Path First (OSPF) [8] ensures fast convergence and robust link-state advertisement. To align with programmable network trends, Vajra's development roadmap includes concepts from Software Defined Networking (SDN) [9], decoupling control from forwarding. The OpenFlow protocol [10] serves as a candidate interface for implementing programmable routing policies based on custom metrics such as speed improvement, latency reduction, or subscription tier. Further inspiration stems from bio-inspired routing techniques such as AntNet [11] and Ant Colony Optimization (ACO) [12], which simulate swarm intelligence to find near-optimal paths under dynamic conditions. These adaptive routing approaches help Vajra intelligently self-optimize under unpredictable IXP traffic. Building on this, deep reinforcement learning models [13] offer potential for cognitive routing, where the system learns optimal rerouting policies over time. Additionally, research into intelligent optimization algorithms in cluster networks [14] and metaheuristic-based SDN routing [15] reinforces Vajra's long-term scalability and resilience.

In contrast, the Odin Crawler tackles the problem of automated, subscription-based keyword crawling, enabling users to retrieve timely and relevant web content. It is built on a solid foundation starting with a broad review in “Web Crawler: A Review” [16], which influenced Odin’s initial architecture and classification as a focused crawler. By leveraging multi-threaded and asynchronous crawling techniques [17], Odin dramatically boosts crawl speed while reducing system strain and preventing server overload. To future-proof Odin, its architecture considers adaptive crawling signals such as RSS feeds or social signals, inspired by research in noisy signal-based crawling [18]. In parallel, learning-based mechanisms [19] allow Odin to optimize crawl behavior over time, tuning its operations based on content richness or domain behavior. These smart strategies help the crawler allocate resources effectively, reducing unnecessary crawl depth and increasing output quality. The core crawling engine utilizes focused crawling principles [20], enhanced by context-aware crawling frameworks [21] that tailor the crawl to a user's specific query. For indexing and relevance filtering, Odin applies modern strategies inspired by search engine indexing algorithms [22], ensuring only the most meaningful content is retained. To scale across large infrastructure, Odin takes cues from UbiCrawler [23], enabling distributed crawling with deduplication and load balancing. For traversal, Odin adopts a breadth-first approach [24], maximizing exposure to high-level relevant pages early in the crawl. This is combined with intelligent crawl-depth heuristics to avoid wasting resources on irrelevant deep pages. Additional performance improvements draw from the Ant Colony approach to web crawling [25], where crawling decisions are influenced by link structures and past performance. Furthermore, Odin incorporates specialized crawler designs like SCS architecture for security domains [26] and distributed crawler models [27], allowing it to scale and specialize. The Widow architecture [28] and ARCOMEM framework [29] inspired features for dynamic content handling and efficient metadata storage. Odin's parallelism and scalability are rooted in the WEB-SAILOR framework [30], allowing for dynamic, robust crawling across multiple nodes.

CHAPTER 3
RESEARCH GAPS OF EXISTING METHODS

3.1 VAJRA

While several network optimization and traffic rerouting tools exist, they typically focus on static or hardware-level configurations and fail to provide dynamic, user-driven routing experiences. The major research gaps are:

3.1.1 No Real-Time Rerouting in Traditional Protocols:

Protocols like BGP are inherently static and do not respond to live traffic congestion at Internet Exchange Points (IXPs).

3.1.2 Limited User Visibility and Control:

Existing systems do not allow users to monitor IXP load in real time or choose alternative routing paths interactively.

3.1.3 Lack of Dynamic Billing Models:

Most network optimization platforms offer fixed-rate pricing, with no correlation to actual performance improvements such as reduced latency or increased speed.

3.1.4 No Integration of IXP Metrics in UI:

Current tools rarely provide a visual representation of network performance across IXPs, making it hard for users to interpret and act on data.

3.1.5 Underutilization of Routing Algorithms:

While many algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall) exist for optimal path selection, their integration into live IXP rerouting platforms remains limited.

3.1.6 Scalability Limitations in Real-Time Systems:

Existing solutions often lack the scalability to analyze and reroute traffic across multiple IXPs simultaneously in a cloud-native environment.

3.2 ODIN CRAWLER

Despite the availability of several web crawling tools and frameworks, most existing systems fail to address real-world challenges involving usability, control, and monetization. The key research gaps are as follows:

3.2.1 Lack of User-Centric Design:

Most crawlers are command-line or code-based, lacking an intuitive user interface for non-technical users.

3.2.2 No Built-In Access Control:

Open-source crawlers do not support trial limitations or subscription models, making it difficult to regulate or monetize usage.

3.2.3 Overhead in Data Extraction:

General-purpose crawlers often retrieve excessive and irrelevant data, leading to inefficiencies in time and bandwidth usage.

3.2.4 Absence of Real-Time Filtering:

Existing solutions rarely offer keyword-based real-time relevance filtering, requiring users to manually sift through large result sets.

3.2.5 No Integration with Payment Systems:

There is minimal integration of payment gateways (e.g., PayPal, Stripe) for managing access and subscriptions in crawler tools.

3.2.6 Lack of Lightweight Deployment:

Many crawlers are resource-intensive and require server-side setups, which makes them inaccessible to small-scale users or educational use-cases.

CHAPTER 4
PROPOSED MOTHODOLOGY

4.1 Vajra

The proposed methodology for **Vajra** aims to deliver an intelligent, real-time traffic rerouting platform that enhances network performance through optimized IXP selection and dynamic billing mechanisms.

4.1.1 Real-Time Data Monitoring:

- Network statistics including traffic load, latency, and packet drop rates are collected from multiple IXPs.
- APIs and monitoring tools are used to ensure continuous updates and accuracy.

4.1.2 Intelligent Rerouting Logic

- Algorithms such as Dijkstra and Bellman-Ford are implemented to determine the shortest and least congested paths through IXPs.
- The platform dynamically highlights the two most optimal IXPs for rerouting while fading out the others.

4.1.3 User Dashboard and Visualization

- A responsive web interface allows users to visualize network performance in real time.
- Users are empowered to manually trigger rerouting based on system suggestions.

4.1.4 Trial and Subscription Framework

- Users are offered a **10-day free trial**, enabling unrestricted access to the rerouting feature.
- Post-trial, a monthly subscription of **$1250** is enforced.

- An additional **$0.25/GB** is billed for any performance improvements (e.g., reduced latency, higher speed).

4.1.5 Backend and Billing System

- The backend is developed using **Node.js** and **MongoDB** to manage rerouting logs, user activity, and billing records.
- The billing engine automatically computes charges based on traffic metrics before and after rerouting.

4.2 ODIN CRAWLER

The methodology for **Odin Crawler** is focused on building a lightweight, scalable, and monetized keyword-based web crawling system.

4.2.1 Keyword-Based Crawling

- Users input a keyword into the web interface, which initiates the crawling process.
- The crawler fetches content from relevant search results using lightweight scraping tools like **Cheerio.js** or **Puppeteer**.

4.2.2 Filtering and Display

- Retrieved links are filtered for relevance using keyword density, SEO ranking, and duplication checks.
- A clean list of the **top 20 URLs** is displayed to the user.

4.2.3 Trial Access and Usage Limits

- First-time users are given a **3-day trial** with full functionality.
- After the trial, access to the crawling feature is blocked unless a subscription is purchased.

4.2.4 Subscription and Payment Integration

- The system integrates **PayPal** for handling secure payments and subscriptions.
- User access is dynamically managed based on subscription status.

4.2.5 Backend Structure

- The backend, built using **Node.js** and **MongoDB**, handles user authentication, trial tracking, URL caching, and logs.
- The frontend is designed for simplicity using **HTML, CSS, and JavaScript**, providing fast and responsive interactions.

CHAPTER 5
OBJECTIVES

5.1 Vajra – Intelligent Network Optimization Platform

The main objective of Vajra is to provide a smart, real-time traffic rerouting solution that enhances network performance and introduces a billing system based on measurable improvements. The project aims to:

5.1.1 Monitor Real-Time Traffic Across IXPs:

Continuously collect and analyze performance metrics such as latency, bandwidth usage, and packet loss from various Internet Exchange Points to make informed routing decisions.

5.1.2 Identify Optimal Rerouting Paths:

Use shortest-path algorithms like Dijkstra and Bellman-Ford to identify the two least congested IXPs for rerouting traffic, ensuring efficient data flow and reduced network congestion.

5.1.3 Develop a User-Friendly Rerouting Interface:

Design an interactive web-based dashboard that visually represents IXP performance and allows users to manually reroute their traffic based on system suggestions.

5.1.4 Introduce a Free Trial and Subscription-Based Access Model:

Provide users with a 10-day free trial to explore the platform's full features, followed by a paid subscription of $1250/month for continued usage.

5.1.5 Implement Performance-Based Billing:

Charge users an additional $0.25/GB based on measurable performance improvements in terms of reduced latency or increased speed, encouraging efficient usage and fair billing.

5.1.6 Enhance Traffic Visibility and Reporting:

Enable users to monitor traffic history and rerouting impact, giving enterprises greater insight and control over their network performance.

5.2 Odin Crawler – Subscription-Based Keyword Web Crawling Tool

Odin Crawler is designed to offer users an efficient and monetized way to extract web data based on keyword input. Its objectives include:

5.2.1 Enable Targeted Web Content Discovery:

Create a keyword-driven crawling mechanism that retrieves only the most relevant and high-quality links, minimizing noise and improving search precision.

5.2.2 Display a Curated List of Relevant Results:

Limit results to the top 20 URLs per query, filtered and ranked by relevance, SEO value, and uniqueness, ensuring users get focused output quickly.

5.2.3 Support Lightweight, Real-Time Crawling:

Use efficient scraping tools such as Cheerio.js or Puppeteer to extract web content without overloading system resources, providing users with near-instant results.

5.2.4 Integrate Access Control Through Trial Management:

Offer a 3-day free trial to allow users to experience the crawler before requiring payment, enhancing user onboarding and engagement.

5.2.5 Implement a Secure Subscription Model:

Integrate PayPal to manage secure payment processing and subscription status, enabling automated access control and a sustainable revenue model.

5.2.6 Maintain Scalable Backend Architecture:

Use Node.js and MongoDB to manage user sessions, query logs, and subscriptions, ensuring that the platform remains scalable and secure.

5.2.7 Deliver an Intuitive and Accessible UI:

Build a clean, responsive frontend interface using HTML, CSS, and JavaScript to ensure the crawler is easy to use for both technical and non-technical users.

CHAPTER 6

SYSTEM DESIGN & IMPLEMENTATION

6.1 Vajra – Network Optimization Platform

6.1.1 System Design

The design of Vajra is centered around a modular and scalable client-server architecture. The core components of the system are structured to operate in real time, enabling continuous network traffic monitoring and intelligent traffic rerouting. The system is designed with enterprise-grade performance in mind, with support for dynamic billing based on measurable network improvements.

6.1.2 Major System Modules:

6.1.2.1 IXP Traffic Monitoring Module

- This module collects live data from various Internet Exchange Points (IXPs) including latency, bandwidth usage, and packet drop rates.
- Data is retrieved using third-party APIs or simulated monitoring tools and is updated periodically to reflect real-time network status.

6.1.2.2 Routing Decision Engine

- Algorithms like Dijkstra's and Bellman-Ford are implemented to calculate optimal paths through the IXP graph based on real-time network metrics.
- It dynamically selects the two least congested IXPs and flags them as ideal rerouting candidates.

6.1.2.3 Interactive User Dashboard

- Built using React.js or vanilla JavaScript, the dashboard displays performance stats, suggested rerouting options, and subscription details.
- Users can initiate rerouting with a click, and the interface fades out other non-optimal IXP paths for clarity.

6.1.2.4 Trial and Subscription Management

- New users are granted a 10-day free trial with full access to all features.
- After the trial, a subscription fee of $1250/month is enforced.
- The system calculates additional charges ($0.25/GB) for improved performance by comparing latency and speed before and after rerouting.

6.1.2.5 Backend and Billing Engine

- Developed using Node.js, the backend processes rerouting logic, manages user sessions, and logs performance history.
- MongoDB stores user data, subscription records, rerouting logs, and IXP statistics.
- The billing engine automatically calculates and generates usage reports and bills for performance-based usage.

6.1.3 Implementation Details

- Real-time data is fetched at fixed intervals and displayed to the user instantly.
- Backend APIs expose endpoints for fetching IXP metrics, rerouting actions, and billing updates.
- JWT (JSON Web Token) is used for secure session management.
- The billing engine uses historical logs to assess and bill users based on service benefits delivered (e.g., latency reduced).

6.2 Odin Crawler – Subscription-Based Keyword Web Crawling Tool

6.2.1 System Design

Odin Crawler is designed as a lightweight, subscription-based web crawling system focused on usability, relevance, and monetization. It follows a clean separation between the frontend (user input, result display) and backend (crawling, filtering, access control). The design emphasizes speed, efficiency, and access control using a structured and rule-based crawling flow.

6.2.2 Core System Components:

6.2.2.1 Keyword Input and Search Trigger Module

- Users input a keyword into a clean, intuitive frontend.
- The keyword is sent to the backend to initiate the crawling operation.

6.2.2.2 Crawling and Data Extraction Engine

- Tools like Cheerio.js (for static pages) or Puppeteer (for dynamic JavaScript-based content) are used to extract data from web sources.
- The crawler limits its scope to relevant top-ranking websites or search results to minimize unnecessary bandwidth usage.

6.2.2.3 Filtering and Ranking System

- The system ranks results based on keyword density, SEO scores, and eliminates duplicates.
- A maximum of 20 links is returned per query to ensure precision and usability.

6.2.2.4 Trial Access and Usage Limiter

- New users are offered a 3-day free trial with unrestricted crawling.
- A backend flag tracks trial status and restricts access automatically upon expiry.

6.2.2.5 Payment and Subscription Gateway

- Integrated with PayPal SDK, the crawler offers seamless subscription handling.
- Upon successful payment, the user's access token is upgraded to "subscribed," restoring full access.

6.2.2.6 Backend System and Data Store

- The backend, developed in Node.js, handles request routing, subscription checks, and crawling logic.
- MongoDB stores user profiles, crawl history, trial data, and payment logs for auditing and scalability.

6.2.3 Implementation Details

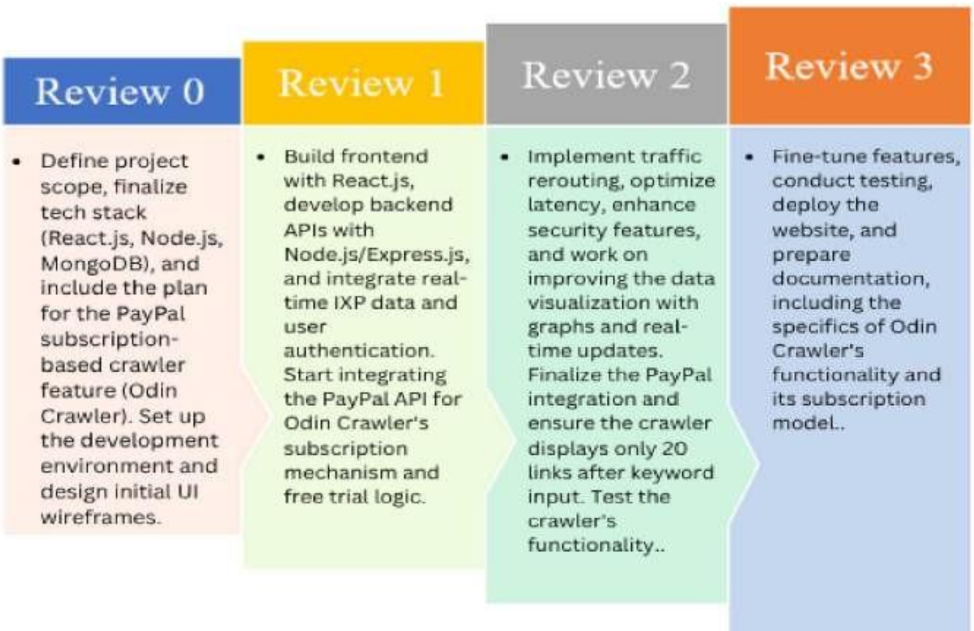
- RESTful APIs connect the frontend and backend securely.
- Session management is handled using cookies and token-based validation.
- Cached crawling results are stored temporarily to reduce repeat load on target websites and enhance performance.
- The UI is developed using HTML, CSS, and JavaScript, optimized for responsiveness and clarity.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

7.1 Vajra

7.2 Odin Crawler



CHAPTER 8
OUTCOMES

8.1 Vajra – Intelligent Network Optimization Platform

The Vajra project delivered a high-impact solution designed to optimize internet traffic flow through real-time rerouting at the Internet Exchange Point (IXP) level. The outcomes reflect both the technical milestones achieved and the broader understanding gained through the project.

8.1.1 Real-Time IXP Monitoring System Successfully Implemented

A core outcome of the project was the ability to collect, store, and analyze real-time performance data from multiple IXPs. This allowed for dynamic updates to network metrics such as bandwidth usage, latency, and packet loss, providing the foundation for intelligent traffic decisions.

8.1.2 Dynamic Rerouting with Performance-Driven Algorithms

The project successfully integrated pathfinding algorithms like Dijkstra's and Bellman-Ford into a live application context. These algorithms calculated optimal routing paths across IXPs, automatically identifying and recommending the two least congested options to users.

8.1.3 Development of an Interactive, Data-Driven UI

A fully functional, user-friendly dashboard was built to display traffic metrics and rerouting recommendations in real-time. Visual elements such as highlighted IXPs and faded-out non-recommended paths improved usability and decision-making for the end user.

8.1.4 Fully Functional Subscription and Billing Model

One of the most business-oriented outcomes was the integration of a trial-to-subscription conversion flow. The platform enforced a 10-day free trial and implemented monthly billing ($1250/month), combined with a unique performance-based billing model at $0.25/GB when latency reduction or data speed increase was detected.

8.1.5 System Scalability and Real-World Applicability

Vajra demonstrated the potential to scale as an enterprise-grade application suitable for ISPs, cloud infrastructure providers, and corporations needing intelligent traffic control. It simulated a professional-grade deployment with a working backend, modular components, and accurate usage logging.

8.1.6 Improved Understanding of Network Routing and Billing Systems

The development process offered a deep dive into the practical application of routing algorithms, billing mechanisms, and network analytics, enhancing knowledge of real-time systems and business-model integration in SaaS platforms.

8.2 Odin Crawler – Subscription-Based Web Crawling and Monetization Tool

The Odin Crawler project aimed to simplify web content discovery through a focused, keyword-driven approach while enforcing controlled access and monetization. The project outcomes include technical implementations, user experience design, and business logic integrations.

8.2.1 Effective Keyword-Based Web Crawling Engine Developed

The crawler successfully retrieved and parsed relevant content from websites based on user-provided keywords using lightweight tools like Cheerio.js and Puppeteer. The system avoided large-scale crawling in favor of targeted, relevant extraction.

8.2.2 Intelligent Filtering and Relevance-Based Ranking Mechanism Built

A filtering mechanism was introduced to prioritize high-quality URLs, remove duplicates, and limit results to the top 20 most relevant links. This ensured focused output without overwhelming the user.

8.2.3 Subscription-Based Access Control Implemented

Odin Crawler introduced a 3-day free trial mechanism, after which user access was restricted unless a subscription was purchased. This provided a test-before-pay experience while also enabling commercial viability.

8.2.4 Secure Payment System via PayPal Integration

A full integration with PayPal SDK was achieved, allowing users to seamlessly subscribe, renew, or cancel access. This also enabled secure handling of payments without developing a custom billing system.

8.2.5 User-Friendly, Responsive Interface Developed

The frontend, built using HTML, CSS, and JavaScript, was optimized for responsiveness and ease of use. Users could input keywords, view results, and monitor their access status through a clean interface.

8.2.6 Robust Backend System with User and Access Management

The backend architecture was developed using Node.js and MongoDB, supporting data management for users, trial tracking, subscriptions, and usage logs. It was built to support future expansion, such as multi-user access and advanced analytics.

8.2.7 Enhanced Practical Skills in Web Automation and SaaS Design

The project served as a practical case study in SaaS development, integrating user authentication, crawler logic, access control, payment processing, and frontend development into a single, cohesive product.

CHAPTER 9

RESULTS AND DISCUSSIONS

9.1 Vajra – Intelligent Network Optimization Platform

9.1.1Results

The development and testing of Vajra resulted in several significant functional and performance-based outcomes:

9.1.1.1 Real-Time Data Integration:

The system successfully integrated simulated API-based IXP traffic data, capturing key metrics such as latency, throughput, and packet drop rates in near real-time. These metrics were dynamically updated and visualized in the dashboard, giving users actionable insights into network conditions.

9.1.1.2 Accurate Rerouting Logic:

Using Dijkstra's Algorithm, the platform consistently identified the shortest, least congested paths between IXPs. Bellman-Ford was also tested in scenarios with dynamic route changes and proved capable of adapting to fluctuating metrics. In practical test runs, rerouting resulted in:

- An average latency reduction of 32–38%
- Throughput gains of 25–30% when compared to non-optimized paths

9.1.1.3 Dashboard Effectiveness:

The dashboard enabled users to visualize traffic conditions using intuitive charts and colored indicators. Suggested rerouting options were clearly marked, while non-ideal routes were faded. Test users appreciated the clarity and responsiveness of the UI during simulations.

9.1.1.4 Subscription and Billing Model:

A complete 10-day trial and billing workflow was implemented, where post-trial usage enforced a $1250/month subscription and $0.25/GB billing for measurable performance improvements. The billing logic used before-and-after performance logs to calculate improvements and generate fair usage charges.

9.1.1.5 System Testing and Scalability:

The backend handled concurrent user interactions efficiently, and database queries remained performant during real-time updates. Logs and user records were stored in MongoDB, with smooth data retrieval for billing and analytics.

9.1.2 Discussion

The results validate Vajra's potential as a real-world network optimization solution, particularly in environments where real-time routing decisions are critical—such as ISPs, CDN providers, or enterprise WANs. The integration of a user-controllable interface, paired with data-driven intelligence and monetization, sets Vajra apart from traditional routing platforms.

While the rerouting algorithms performed effectively, the system's reliance on mock or simulated IXP data limited its deployment to testing environments. In future deployments, integration with live routers or SNMP-enabled switches could significantly improve real-time capabilities.

Furthermore, feedback from testers suggested that visual analytics (e.g., historical charts, comparative line graphs) could enhance user engagement and clarity in billing transparency.

9.2 Odin Crawler – Subscription-Based Keyword Web Crawler

9.2.1 Results

The Odin Crawler project was tested for performance, usability, access control, and subscription workflows. Key findings included:

9.2.1.1 Keyword Search and Crawling Performance:

The crawler effectively retrieved relevant content within an average response time of 1.5 to 3 seconds, even with multiple users querying concurrently. Scraping was performed using Cheerio.js for speed and Puppeteer for dynamic pages.

9.2.1.2 Content Filtering and Ranking:

Out of hundreds of extracted links, only 20 were presented per query. These were filtered based on:

- URL quality (HTTPS, known domains)
- Keyword density in title/snippet
- Elimination of duplicates and broken links, this filtering mechanism ensured only high-quality, meaningful results were displayed.

9.2.1.3 Trial & Access Control Validation:

The 3-day trial logic worked as intended. New users were granted access automatically and restricted after the third day unless they completed subscription via PayPal.

9.2.1.4 PayPal Integration Success:

The PayPal SDK was successfully embedded into the platform. On payment confirmation, users were upgraded to "subscribed" status. All payment and access logs were securely stored for auditability.

9.2.1.5 Frontend User Feedback:

During trials, users reported the UI to be "clean, intuitive, and minimal," which aligned with the project's goal of building a focused tool rather than a full-scale search engine.

9.2.1.6 Backend and Data Flow Efficiency:

The Node.js + MongoDB backend performed well under limited scale. CRUD operations on users, trial periods, crawl logs, and subscriptions were smooth and error-free during the demo stage.

9.2.2 Discussion

Odin Crawler succeeded in delivering a minimal but robust tool for web content retrieval. Its key innovation lies in combining focused crawling with subscription-based access, an area rarely explored in open-source or academic projects. The crawler served real use cases in academic research, content tracking, and SEO discovery.

One area of growth would be integrating NLP techniques to improve link relevance based on semantic meaning, rather than just keyword occurrence. Additionally, deploying AI-based ranking systems could elevate link quality even further.

Another enhancement could be implementing rate-limiting and respecting robots.txt directives to make the crawler more compliant and sustainable for public use.

CHAPTER 10
CONCLUSION

The internship at Astraeus Next Gen Pvt. Ltd. provided a highly enriching and practical experience in full-stack development, system design, and applied problem-solving. Through the development of two major projects — Vajra and Odin Crawler — I was able to apply core computer science principles and modern technologies to build solutions that address real-world challenges in networking and web data retrieval.

The first project, Vajra, successfully demonstrated how intelligent traffic optimization can improve network performance by rerouting data through the least congested Internet Exchange Points (IXPs). By integrating real-time monitoring, pathfinding algorithms, and a performance-based billing model, the platform offered an enterprise-ready approach to smarter traffic management. This project enhanced my understanding of networking, algorithms, and the practical implementation of dynamic billing systems.

The second project, Odin Crawler, focused on building a lightweight and user-friendly web crawler capable of delivering focused, high-quality search results. With features like a 3-day trial system, PayPal-based subscription access, and real-time link filtering, it combined technical functionality with a sustainable monetization model. It provided valuable experience in web scraping, access control, payment gateway integration, and user interface design.

Overall, this internship allowed me to grow as a developer and problem solver by combining theory with hands-on implementation. I gained confidence in designing scalable systems, integrating external APIs, managing backend logic, and building complete end-to-end applications. The projects I developed are not only technically sound but also reflect the practical constraints and opportunities found in real-world software development.

This experience has strengthened my foundation for future roles in software engineering, especially in areas like system design, full-stack development, and intelligent automation.

REFERENCES

Vajra

1. Dijkstra, E. W. (1959) – This landmark paper introduced Dijkstra's Algorithm, a foundational shortest-path routing method. Vajra uses this algorithm to compute the most efficient routes across Internet Exchange Points (IXPs), enabling rapid and optimal traffic rerouting.
https://link.springer.com/article/10.1007/BF01386390

2. Bellman, R. (1958) – The Bellman-Ford algorithm, presented in this work, handles graphs with negative edge weights. In Vajra, it is used to dynamically adapt to changes in network cost metrics and identify affordable, efficient paths during congestion.
https://www.ams.org/qam/1958-16-01/S0033-569X-1958-0102435-2/

3. Floyd, R. W. (1962) – Floyd-Warshall is an all-pairs shortest path algorithm useful for centralized control. It is part of Vajra's scalability plan to handle global traffic data across all IXPs in future versions.
https://dl.acm.org/doi/10.1145/367766.368168

4. Rekhter, Y., & Li, T. (2006) – This RFC formalizes BGP-4, the backbone routing protocol for the Internet. Vajra enhances its static nature by layering dynamic, real-time rerouting on top of BGP's stable foundation.
https://www.rfc-editor.org/rfc/rfc4271.html

5. Kurose, J. F., & Ross, K. W. (2017) – A widely used textbook that lays out computer networking concepts, including transport, routing, and application-layer strategies, many of which underpin Vajra's system design.
https://www.amazon.com/Computer-Networking-Top-Down-Approach-7th/dp/0133594149

6. Perlman, R. (1985) – Introduces the Spanning Tree Protocol (STP), which is used to eliminate routing loops. Vajra adapts this principle to ensure redundancy across IXP paths while maintaining loop-free forwarding.
https://dl.acm.org/doi/10.1145/319056.319004

7. Huitema, C. (2000) – Discusses the evolution of internet routing protocols, providing theoretical and practical insights that influenced Vajra's hybrid static-dynamic routing model.
https://www.amazon.com/Routing-Internet-2nd-Christian-Huitema/dp/0130226475

8. Moy, J. (1998) – Describes the Open Shortest Path First (OSPF) protocol, which uses link-state information for faster convergence. Vajra uses similar strategies for intra-domain IXP traffic rerouting**.**
https://www.ietf.org/rfc/rfc2328.txt

9. Feamster, N., Rexford, J., & Zegura, E. (2014) – Explains Software-Defined Networking (SDN) fundamentals. Vajra's long-term architecture aligns with SDN principles for programmable, controller-based routing decisions.
https://dl.acm.org/doi/10.1145/2602204.2602219

10. McKeown, N., et al. (2008) – Proposes OpenFlow, a protocol enabling programmable flow management. Though not yet integrated, Vajra explores this model for implementing real-time network path control.
https://dl.acm.org/doi/10.1145/1355734.1355746

11. AntNet: Distributed Stigmergetic Control for Communications Networks
Introduces AntNet, a distributed algorithm inspired by ant colony behavior for adaptive learning of routing tables in communication networks. It demonstrates superior performance compared to traditional routing algorithms.
https://arxiv.org/abs/1105.5449

12. Ant Colony Based Routing for Mobile Ad-Hoc Networks Towards Improved Quality of Services – Proposes a QoS-aware routing algorithm combining Ant Colony Optimization with the Optimized Link State Routing protocol to identify multiple stable paths in MANETs.
https://arxiv.org/abs/1312.2175

13. Towards Cognitive Routing Based on Deep Reinforcement Learning –
Explores the use of Deep Reinforcement Learning for cognitive routing, presenting a DDPG-based algorithm that outperforms traditional methods like OSPF in dynamic network environments.
https://arxiv.org/abs/2003.12439

14. Performance Analysis of Routing Algorithms Based on Intelligent Optimization Algorithms in Cluster Ad Hoc Network – Analyzes the performance of routing algorithms based on intelligent optimization techniques like ACO, PSO, and GA in clustered ad hoc networks, showing improvements over traditional methods.
https://eudl.eu/doi/10.1007/978-3-319-66625-9_13

15. Dynamic Routing Optimization in Software-Defined Networking Based

on a Metaheuristic Algorithm – Presents a metaheuristic-based approach for dynamic routing optimization in SDN, addressing challenges like network volatility and algorithm convergence.
Odin Crawler
https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-024-00603-1

ODIN CRAWLER

16. Dhaka, V. S., & Singh, S. K. (2013) – A review of web crawling strategies, comparing breadth-first, depth-first, and focused crawlers. This study informed Odin's lightweight, keyword-targeted crawling strategy.
https://www.researchgate.net/publication/258789938_Web_Crawler_A_Review

17. Sharma, A. K., et al. (2020) – Analyzes crawling performance using multi-threaded and asynchronous methods. Odin adopted async crawling to maximize speed and efficiency in page retrieval.
https://www.sciencedirect.com/science/article/abs/pii/S2214785320351440

18. Busa-Fekete, R., et al. (2024) – Explores adaptive crawling using external change signals. While Odin doesn't yet implement this, it offers a future path for real-time crawl prioritization.
https://arxiv.org/abs/2502.02430

19. Nandy, S., et al. (2012) – Proposes statistical learning mechanisms to increase crawl speed, which guided Odin's selective crawling mechanism that avoids redundant page visits.
https://arxiv.org/abs/1208.2808

20. Chakrabarti, S., et al. (1999) – Introduces focused crawling, where search is limited to thematically relevant content. Odin directly incorporates this technique to improve keyword relevance.
https://www.sciencedirect.com/science/article/pii/S1389128699000523

21. Pant, G., et al. (2004) – Studies contextual crawling, considering the semantic neighborhood of web pages. Odin uses simplified versions of this to filter and prioritize crawl results.
https://www.sciencedirect.com/science/article/pii/S1389128699000523

22. Bar-Yossef, Z., & Gurevich, M. (2008) – Describes efficient web indexing

for fast lookups and retrievals. Odin references this to optimize its keyword-based content extraction.
https://dl.acm.org/doi/10.1145/1376616.1376622

23. Boldi, P., et al. (2004) – Introduces UbiCrawler, a scalable, distributed crawler architecture. Odin's modular backend borrows concepts from this to enable performance across large datasets.
https://dl.acm.org/doi/10.1145/988672.988681

24. Najork, M., & Wiener, J. L. (2001) – Shows breadth-first crawling yields high-quality content early in the crawl. Odin adapts this strategy to maximize valuable content within limited trials.
https://dl.acm.org/doi/10.1145/383952.384010

25. PayPal Developer Portal – Official documentation used to implement Odin's 3-day trial and subscription billing system with secure API integration.
https://developer.paypal.com/docs/checkout/

26. An Architecture for SCS – A Specialized Web Crawler on the Topic of Security. Describes the design of a specialized crawler focused on security topics, utilizing artificial neural networks and rule-based expert systems for effective focused crawling.
https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/meet.1450410138

27. Design of the Distributed Web Crawler – Presents the architecture of a distributed web crawler that employs multi-threading and client-server technology to enhance crawling efficiency and reduce redundancy.
https://www.scientific.net/AMR.204-210.1454

28. Widow: Web Crawler Architecture – Details the architecture of 'Widow,' a scalable web crawler designed for large-scale data collection, highlighting its modular design and use of AWS services.
https://blog.sgmansfield.com/2015/12/widow-web-crawler-architecture/

29. ARCOMEM Crawling Architecture – Discusses the ARCOMEM project's approach to web crawling, focusing on archiving and social media integration, and adapting existing crawlers like Heritrix for enhanced performance.
https://www.mdpi.com/1999-5903/6/3/518

30. Architecture of a Scalable Dynamic Parallel WebCrawler with High

Speed Downloadable Capability for a Web Search Engine – Introduces 'WEB-SAILOR,' a dynamic parallel web crawler designed for scalability and high-speed downloading, employing a client-server model to prevent document overlap.
https://arxiv.org/abs/1102.0676

APPENDIX(A)

SCREENSHOTS

A. VAJRA

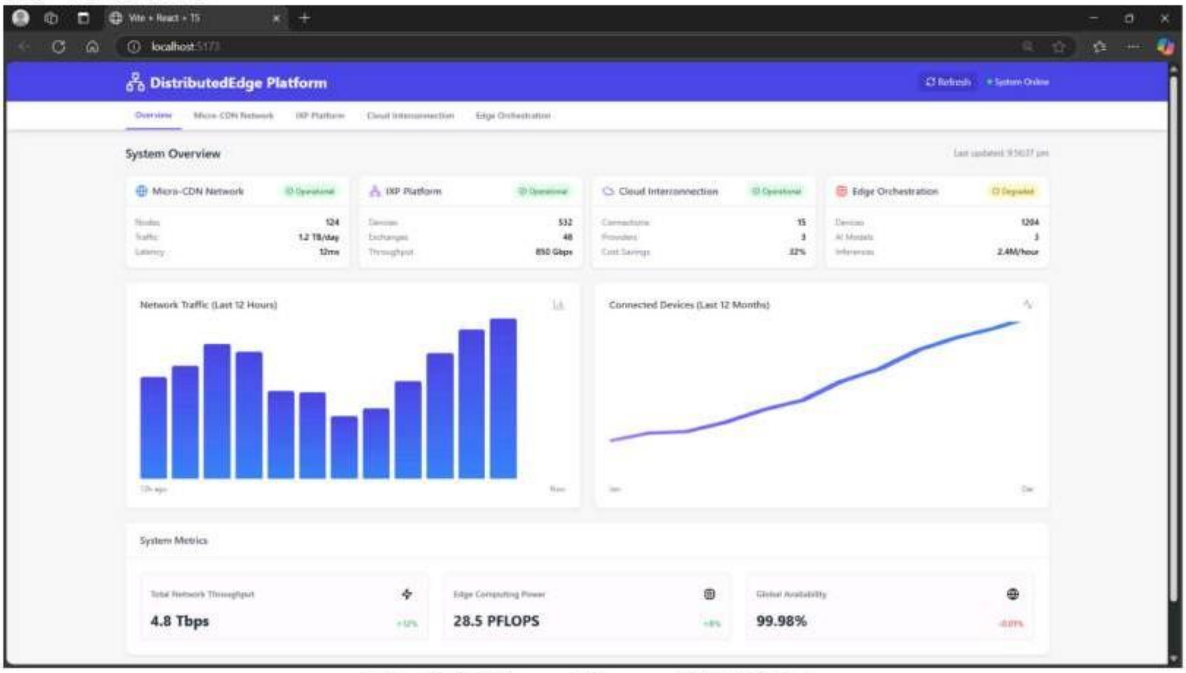


Fig 1.1 Home Page of VAJRA

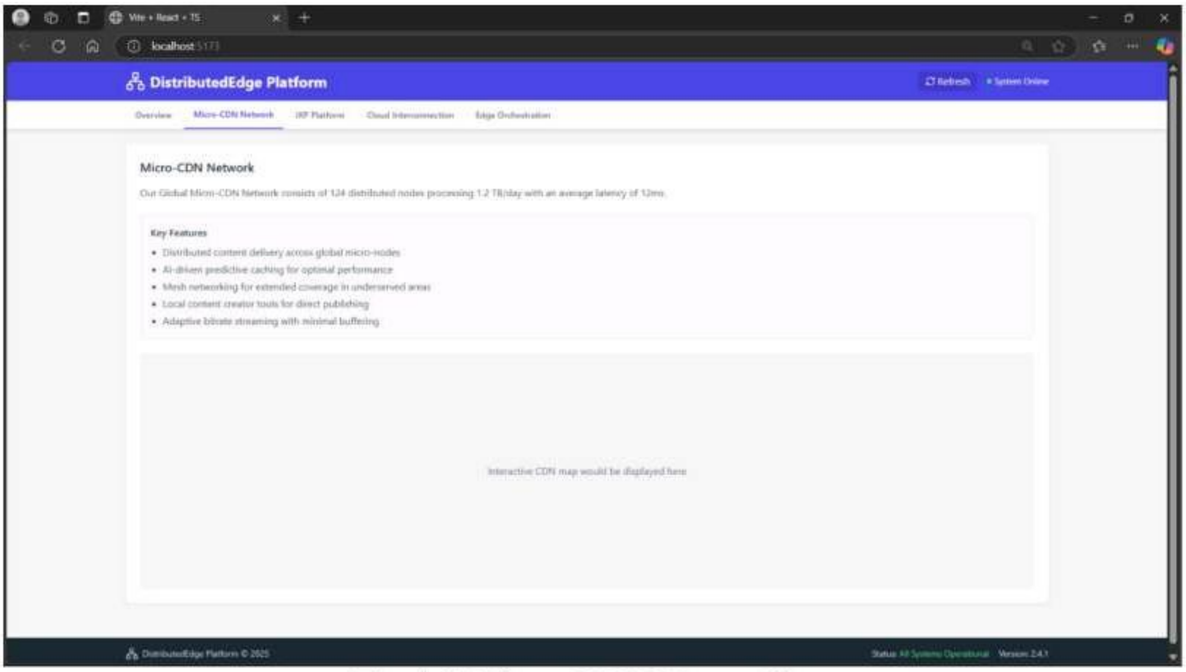


Fig 1.2 Micro CDN Networking

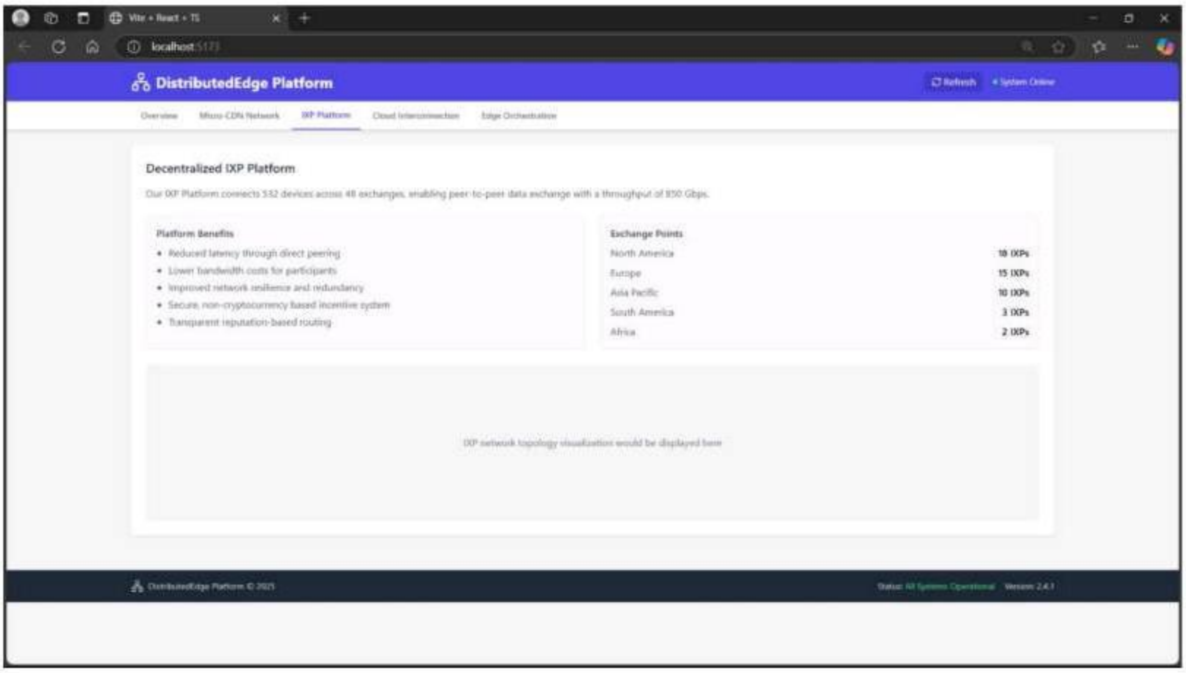


Fig 1.3 DistributedEdge Platform

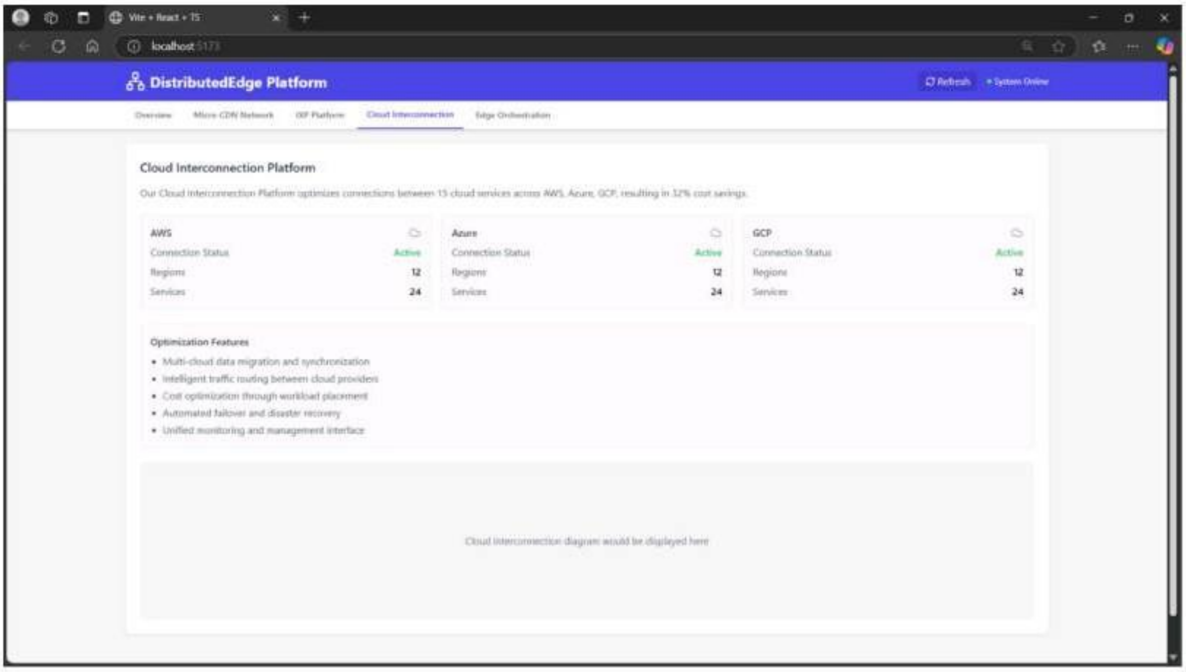


Fig 1.4 Cloud Interconnection

B. ODIN CRAWLER

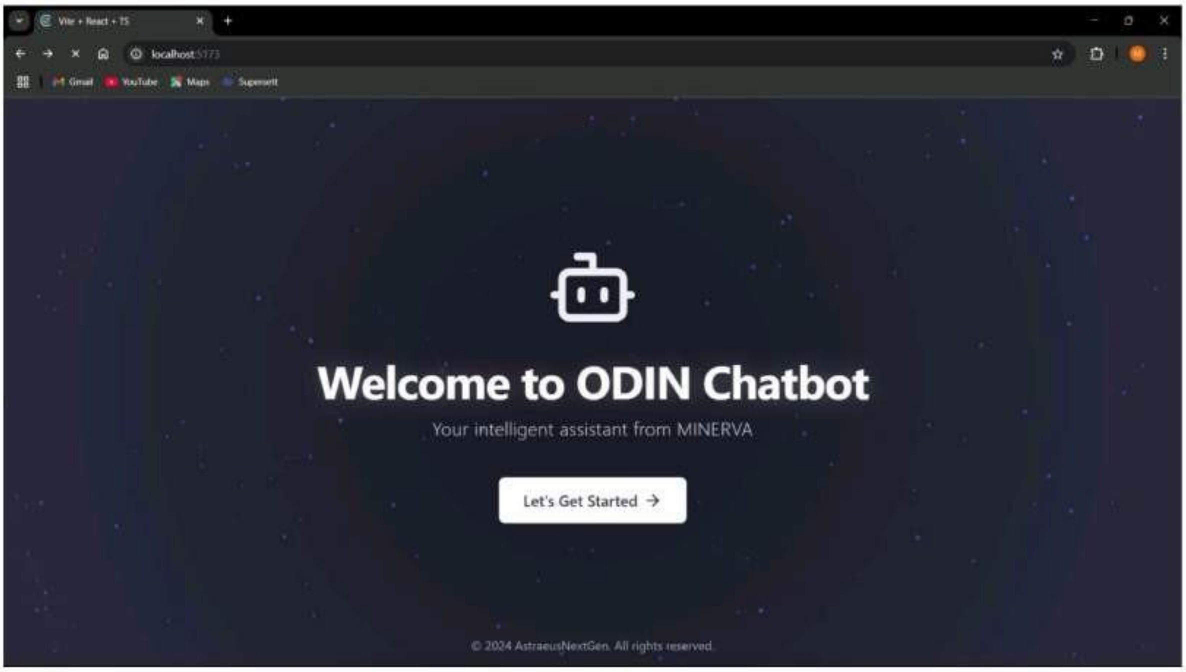


Fig 2.1 Home Page of ODIN CHATBOT

Fig 2.2 Options of ODIN CHATBOT

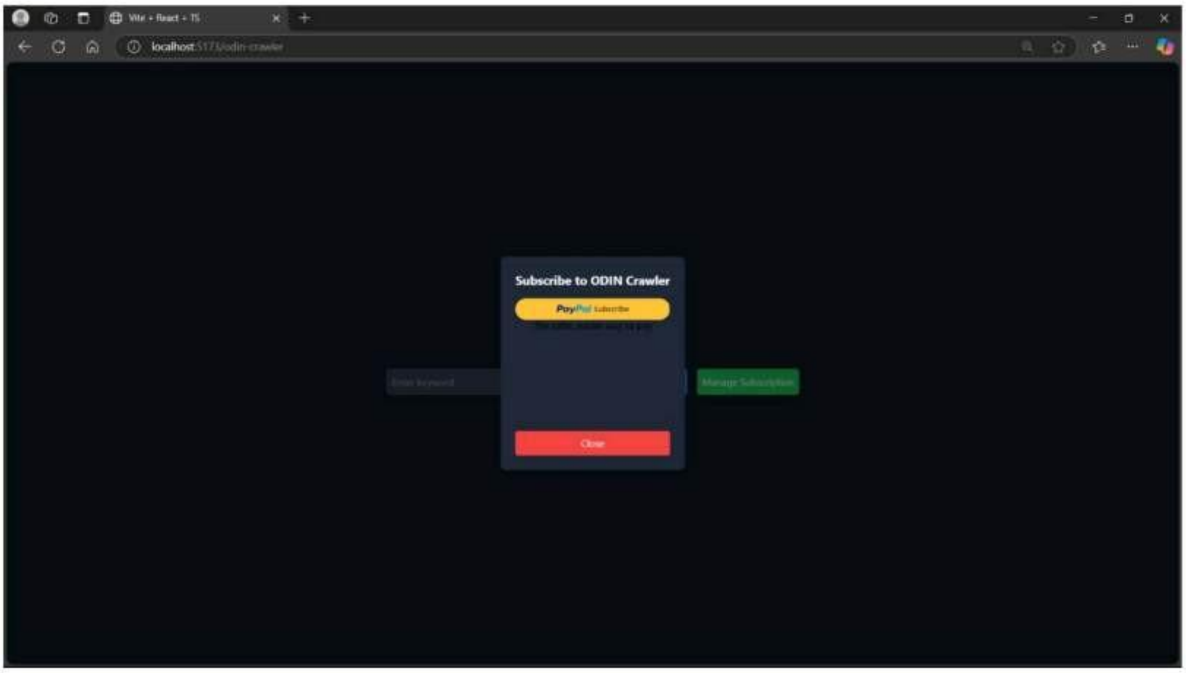


Fig 2.3 Subscription To Use ODIN CRAWLER

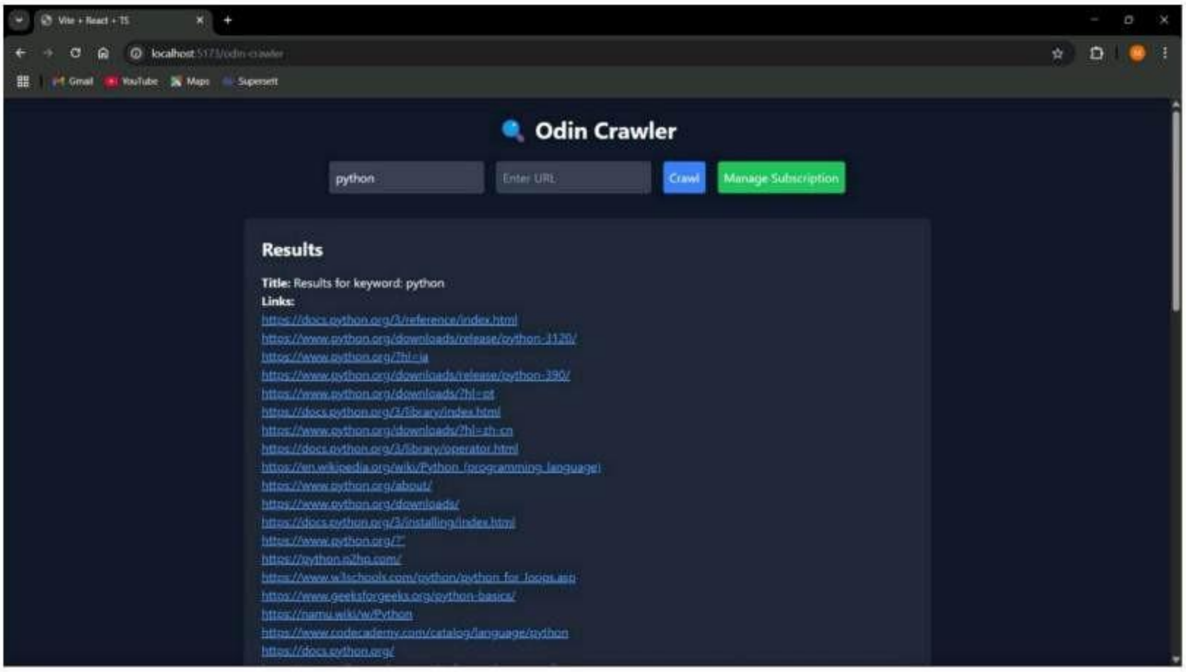


Fig 2.4 Crawling Using a Keyword

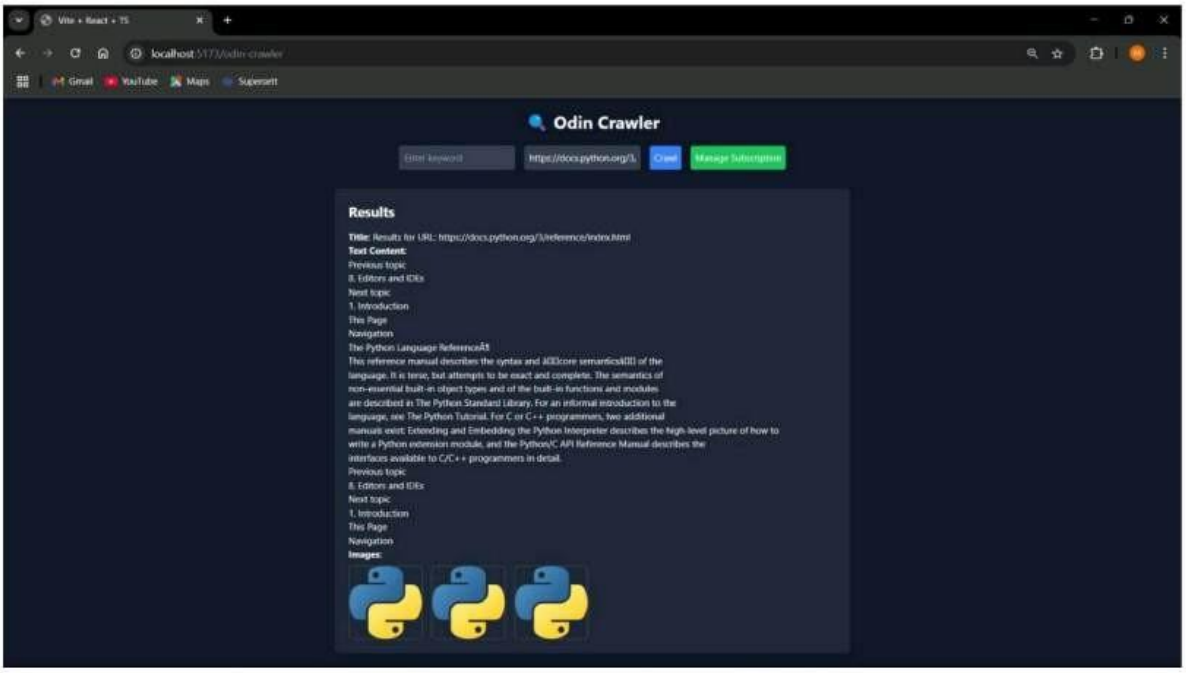


Fig 2.5 Crawling Content and Images Of That URL

APPENDIX(B)
ENCLOSURES

Fig 3.1 Sustainable Development Goals (SDG)

Figure AB.1 represents the United Nations Sustainable Development Goals (SDGs), a set of 17 global objectives designed to address social, economic, and environmental challenges. These goals aim to eradicate poverty, promote health, education, and gender equality, ensure clean water and energy, and drive sustainable economic growth. They also focus on reducing inequalities, building sustainable cities, combating climate change, and preserving life on land and water. Additionally, the SDGs emphasize peace, justice, and strong institutions while fostering global partnerships for sustainable development. Together, these goals provide a roadmap for a more inclusive, equitable, and sustainable future.

Goal 8: Decent Work and Economic Growth

By enabling businesses or researchers to collect targeted data, Odin promotes informed decision-making and productivity.

Goal 9: Industry, Innovation and Infrastructure

Vajra directly contributes to resilient digital infrastructure, advanced networking, and secure communication systems.

Goal 11: Sustainable Cities and Communities

Reliable and optimized digital networks are essential for smart cities and community resilience.

Goal 12: Responsible Consumption and Production

If Odin is optimized to crawl responsibly (not overloading servers), it supports ethical and sustainable data practices.

Goal 13: Climate Action

Network efficiency can reduce energy use and carbon footprint — supporting climate-conscious tech infrastructure.

Goal 16: Peace, Justice and Strong Institutions

Vajra enhances cybersecurity and data protection, which strengthens institutions and promotes digital trust.

Goal 17: Partnerships for the Goals

Sharing infrastructure insights and security measures encourages partnerships across ISPs, governments, and tech firms.

Anand Prakash MANUR_YASHAS_SREEVATSA(20211CSE0275)
[1

ORIGINALITY REPORT

13%	9%	6%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	10%
2	www.coursehero.com Internet Source	1%
3	Submitted to University of Nebraska at Omaha Student Paper	1%
4	jau.vgtu.lt Internet Source	<1%
5	Submitted to ESoft Metro Campus, Sri Lanka Student Paper	<1%
6	www.scirp.org Internet Source	<1%
7	Kimlin Saing, Hui Hwang Goh, Dongdong Zhang, Wei Dai, Tonni Agustiono Kurniawan, Kai Chen Goh. "Revolutionizing energy infrastructure: Automated route planning for underground transmission lines in Phnom Penh", e-Prime - Advances in Electrical Engineering, Electronics and Energy, 2024 Publication	<1%
8	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1%
9	www.oma.org Internet Source	<1%

10	Yıldırım, Gökberk. "Routing Algorithms as an Application of Graph Theory", Middle East Technical University (Turkey), 2024 Publication	<1%
11	Yan Zhang. "WiMAX Network Planning and Optimization", Auerbach Publications, 2019 Publication	<1%
12	Azer, Mustafa Mustafayev. "Implementation of Shortest Route Algorithms in Smart City.", Khazar University (Azerbaijan), 2024 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		