Student Project proposal

*of*

**Namma Salai Backend Enhancement: Server Migration,**

**API Update and Database Modernization**

***Submitted by***

***Namasivayam.M***

***Adhithyan***

***Silambu***

***Saravana Kumar***

*to*

# The Member Secretary,

**Tamilnadu State Council for Science and Technology, DOTE Campus,**

# Chennai - 600 025

## INDEX

|  |  |  |
| --- | --- | --- |
| **S. No** | **Content** | **Page No** |
| **1** | **Project proposal General information** | **1** |
| **2** | **Annexure – 1 Project Details**  Namma Salai Backend Enhancement: Server Migration, API Update, and Database Modernization | **2** |
| **3** | **Annexure – 2 Bonafide Certificate** | **8** |
| i. NAMASIVAYAM.M |
| ii. ADHITHYAN | **9** |
| iii. SILAMBU | **10** |
| iv. SARAVANA KUMAR | **11** |

**PROJECT PROPOSAL**

1. **Name of the Students :** NAMASIVAYAM.M ([iamnamachu40@gmail.com](mailto:iamnamachu40@gmail.com))

ADHITHYAN

SILAMBU

SARAVANA KUMAR

1. **Name of the Guide :** JEYA LAKSHMI

**Department / Designation :** Computer Science and Engineering

Institutional Phone No & Address  **:** **MANGAYARKARASI COLLEGE OF ENGINEERING**

**PARAVAI, MADURAI - 625402**

**Approved by AICTE, New Delhi**

**& Affiliated to Anna University,**

**Chennai-600 025**

**PHONE:**[0452 266 8635](file:///C:\\Users\\MR%20GREEN%20YT\\Downloads\\0452%20266%208635.docx)

[Mail to:mangai.enggcoll@gmail.com](mailto:mangai.enggcoll@gmail.com)

### Project Title Namma Salai Backend Enhancement: Server Migration, API Update, and Database Modernization

1. **Sector in which your Project**

**proposal can be carried out :** Engineering & Technology

### Project Details : Refer annexure - 1

1. **Has a similar project been carried : Out in your college / elsewhere? If so furnish details of the previous project and highlight the improvements suggested in the present one**

**Yes :** The application "Namma Salai" is already operational but faces significant challenges with its backend infrastructure and database management.

### Highlights :

* Revamping the application’s API to incorporate modern standards and practices, **improving functionality, security, and performance.**
* Migrating the application to a new server environment to enhance **scalability, reliability, and overall efficiency**.
* Overhauling the existing database system to a more **robust and efficient platform**, addressing issues related to **data management and performance.**

### Annexure – 1 Project Details

**Namma Salai Backend Enhancement: Server Migration, API Update, and Database Modernization**

**Introduction :**

* **Problem**: The existing "Namma Salai" software, whilst functional, faces numerous backend-associated challenges that impact its overall performance and scalability. The key problems encompass an old server infrastructure, inefficient API design, and a database system that struggles with records consistency and performance. These barriers restrict the utility's ability to handle growing person needs and affect overall consumer delight.
* **Solution**: To cope with those challenges, the mission focuses on a complete backend enhancement for "Namma Salai." This consists of migrating to a more superior server surroundings, updating and optimizing the API structure, and modernizing the database system. The goal is to improve overall performance, scalability, and reliability, ensuring a stronger and efficient backend infrastructure.

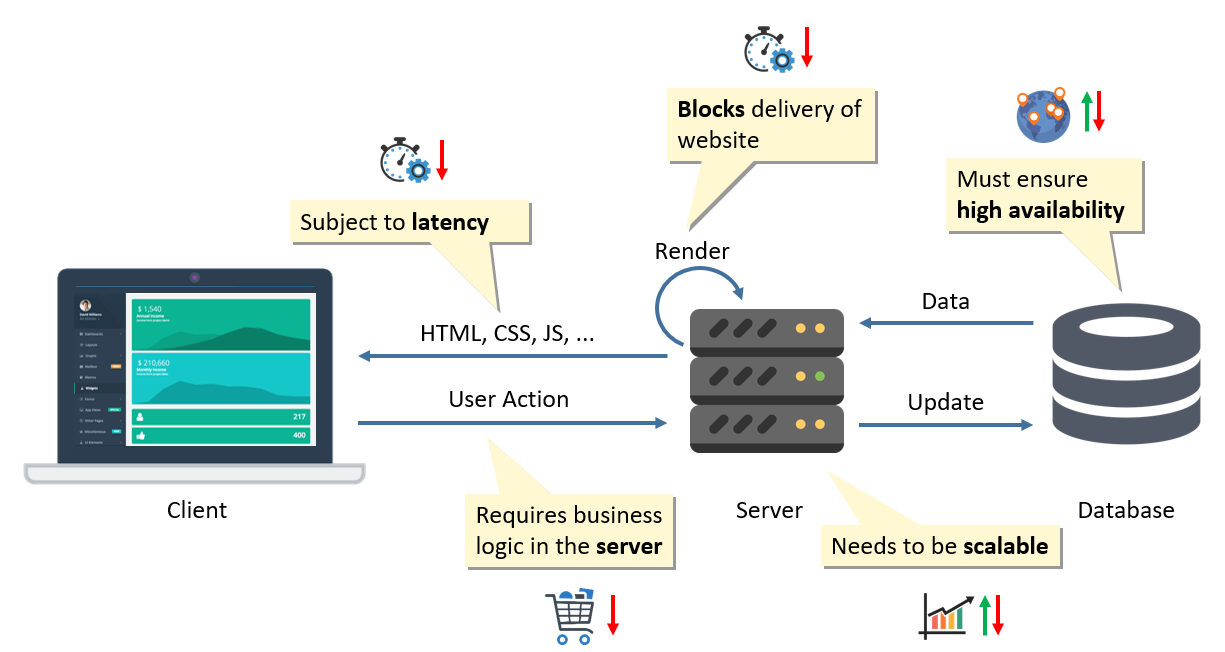
**Scope:**

* Move the existing backend server infrastructure to a new environment to improve scalability, reliability, and performance.
* Update and enhance the existing APIs to improve functionality, security, and performance.
* Upgrade and modernize the database to enhance performance, scalability, and data management capabilities.

**Objective:**

* The primary objectives of my project is to modernize the existing backend by updating the API to meet contemporary standards for improved functionality and security, migrating to a new server environment to enhance scalability and performance, and modernizing the database system to increase efficiency, data management capabilities, and retrieval speed.

**Methodology:**

****

### Research on “Namma Salai” app backend:

### Current System Analysis: Review existing backend systems, APIs, and databases for specific issues and needs.

### Requirements Gathering: Collaborate with stakeholders to gather detailed requirements for new servers, APIs and database systems.

### Project planning: Create a detailed project plan that outlines timelines, resources and milestones.

### Redesign API:

### The core of the redesign process is to reimagine the API’s structure and capabilities. One of the fundamental decisions is whether to maintain a RESTful architecture or adopt newer paradigms like GraphQL. REST is widely used and understood, but GraphQL offers more flexible and efficient data querying capabilities, which may be beneficial for certain applications.

### Performance improvements are central to the redesign. Introducing caching mechanisms can significantly enhance response times by reducing the need to repeatedly fetch data from the database. Implementing pagination and filtering helps manage large datasets more effectively, providing users with a more responsive experience. Additionally, incorporating asynchronous processing for long-running operations can prevent blocking and improve overall throughput.

### Performance testing is essential to ensure that the API can handle expected traffic and perform optimally under various load conditions. This helps identify any potential bottlenecks and allows for adjustments to be made before the final release.

### Capture.PNG

### . Enhancing the Backend Server :

### In today’s digital landscape, the performance and reliability of an application’s backend server are critical to its success. For an application like "Namma Salai," which likely serves a significant user base and manages substantial data, enhancing the backend server is essential to meet growing demands and ensure a robust user experience.

### Implementing load balancing distributes incoming traffic across multiple server instances. This prevents any single server from becoming a bottleneck, improving the application's responsiveness and uptime.

### Integrating caching mechanisms can significantly reduce server load and improve response times. By caching frequently accessed data in memory, such as user profiles or popular content, the server can serve requests more quickly and reduce the need for repeated database queries.

### Optimizing database performance is crucial for backend efficiency. This includes indexing frequently queried fields, optimizing database queries, and regularly maintaining the database through cleaning and defragmentation.

### Capture2.PNG

### Technology Used:

### Database Optimization:

### MySQL 8.0

### ****Microservices Architecture:****

**Kong 3.0** and **NGINX 1.24**

* **Load Balancer:**

### HAProxy 2.7

### ****Containerization and Orchestration:****

### Docker 24.0

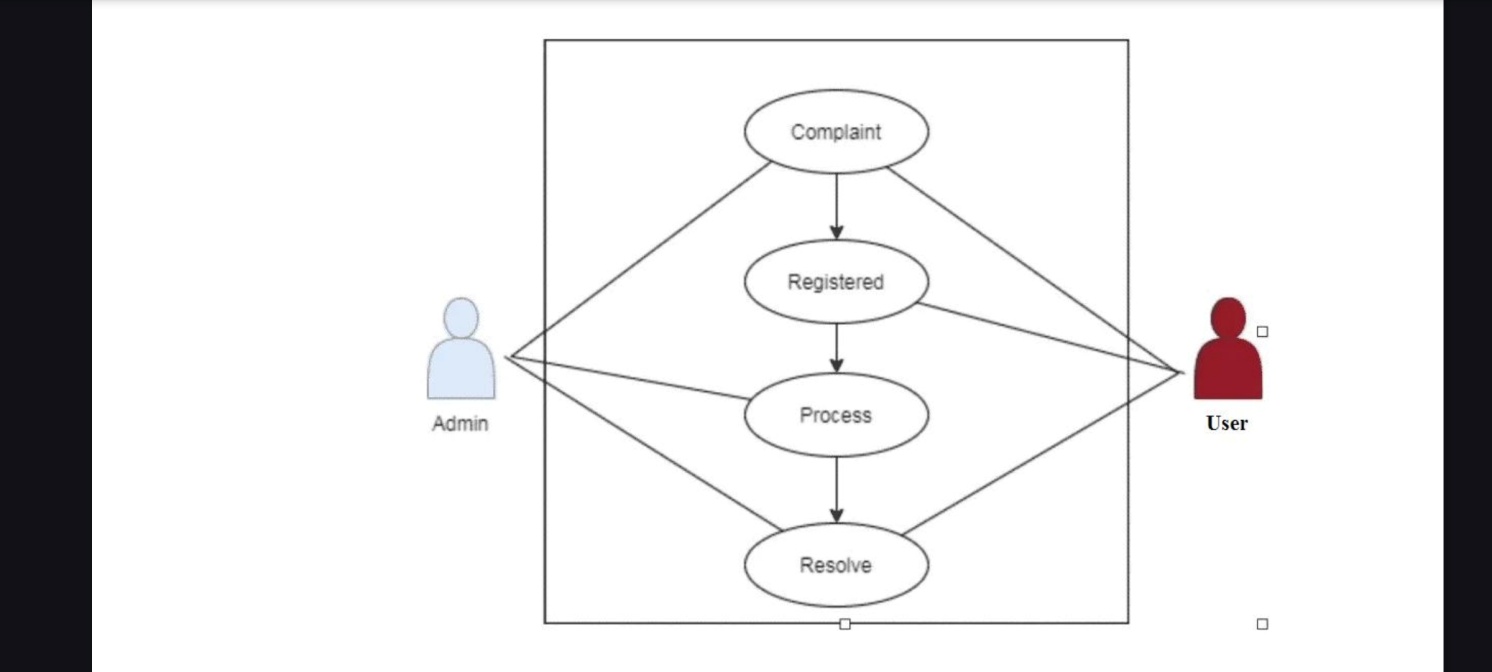
### API Management:

### Kong Enterprise 3.0 and Apigee 2024

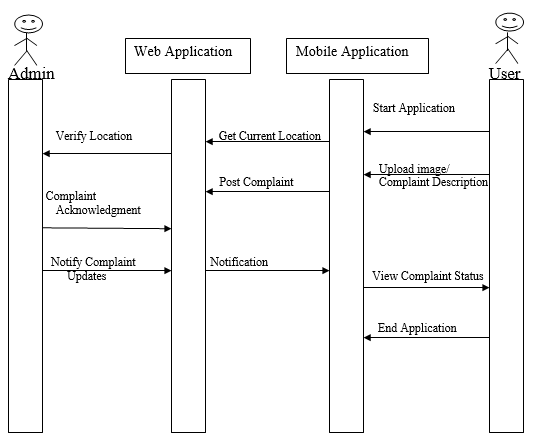
### ****Monitoring and Logging:****

### Prometheus 2.45, Grafana 9.6 and ELK Stack

### 

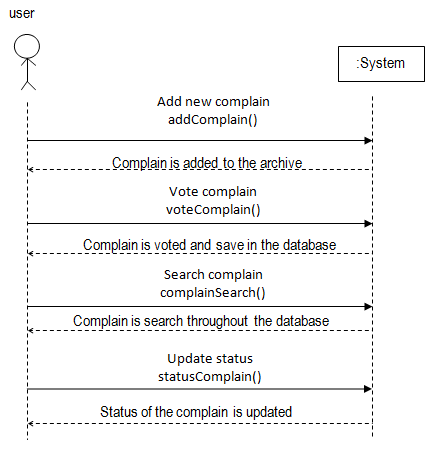


### Use Case Diagram



**Activity Diagram**

### Sequence Diagram



### ****Budget****

* **Server Migration Costs**: 6,000 inr
* **API Update Costs**: 2,500 inr
* **Database Modernization Costs**: 2,000

**Total Budget:** 10,500 inr

### ****Timeline****

* **Phase 1: Server Migration** – 2 week
* **Phase 2: API Update** – 3 week
* **Phase 3: Database Modernization** – 3 week

**Overall Project Duration:** 8 week

**Conclusion:**

* The backend infrastructure's performance has been significantly improved, leading to faster load times and more responsive application behavior.
* The new server and database environments provide a scalable foundation that can accommodate increased traffic and data growth with minimal disruption.
* Updated APIs and modernized database systems incorporate advanced security measures, reducing vulnerabilities and protecting sensitive data.
* The updated infrastructure is easier to maintain, with improved documentation and streamlined processes, reducing the complexity of future updates and enhancements.