

|  |
| --- |
| e-KHANA  2024 |
|  |
| September 8  Skill Spark  Authored By: Pratik Devkota Submitted To: Ram Udgar Yadav |

**Table of Content**

[1. Project Selection Rationale: 3](#_Toc176642456)

[1.1. Why I Chose the Project: 3](#_Toc176642457)

[1.2. Target Audience: 3](#_Toc176642458)

[1.3. Core Features 3](#_Toc176642459)

[2. Technical Justification: 4](#_Toc176642460)

[2.1. Technologies, Tools, and Frameworks 4](#_Toc176642461)

[2.2. Schema Design Considerations 4](#_Toc176642462)

[2.3 User Interactions 4](#_Toc176642463)

[3. Expected Challenges and Solutions 5](#_Toc176642464)

[3.1. Potential Challenges 5](#_Toc176642465)

[3.2. Proposed Solutions 5](#_Toc176642466)

[4. Expected Outcomes and Benefits 5](#_Toc176642467)

[4.1. Impact on Target Audience 5](#_Toc176642468)

[4.2 Learning Objectives 5](#_Toc176642469)

[5. Conclusion 6](#_Toc176642470)

# **Project Selection Rationale:**

## ****Why I Chose the Project****:

The decision to create Ekhana, an online food delivery platform, stems from my personal interest in web development and the growing significance of online food delivery services. Specially, on Nepal where business in food industry seems to be bigger so there might be demand of this project.

This platform aligns well with real-world needs, particularly in urban areas where convenience and time-saving are prioritized. The project is expected to present challenges in terms of ensuring scalability, seamless user experience, and integrating real-time features such as order tracking. Tackling these challenges will be a valuable learning experience, enhancing both my technical skills in web development and understanding of complex systems.

## **Target Audience:**

Ekhana will primarily give services to:

**Urban Residents**: Professionals and families seeking convenience in food delivery.

**Karnali Province:** Since there aren't many food delivery websites currently, I believe the demand will grow in the near future. I aim to offer this service to support my community.

**Restaurants**: Local eateries that want to expand their reach without investing in their own delivery infrastructure.

**Students and Working Professionals**: Those who need quick, reliable meal options, especially during busy schedules.

**Tourists**: People unfamiliar with local restaurants but needing a fast, easy solution for meals.

## ****Core Features****

**User-Friendly Interface**: A clean, responsive design that makes it easy for users to browse menus, select food, and complete orders.

**Real-Time Order Tracking**: Track the status of food from preparation to delivery, offering transparency and reassurance.

**User Authentication**: Secure login and sign-up functionality for both customers and restaurant owners.

**Restaurant Listings & Menus**: A categorized system that allows users to view various restaurants and their food options, with filters for cuisine, price, and delivery time.

**[optional]Payment Gateway Integration**: Secure online payments via credit card, debit card, and mobile wallets.

**Reviews and Ratings**: A 5-star rating system allowing customers to rate both restaurants and delivery service.

**ChatBot**: thinkin g of creating a chat bot so they can directly communicate with resturant and admin.

**[optional]Admin Dashboard**: A control panel for managing users, orders, and restaurant listings.

These features are essential because they create a seamless user experience, enhance trust in the platform, and ensure smooth operational management.

# 2. **Technical** **Justification**:

## ****2.1. Technologies**, **Tools**, **and** **Frameworks****

**MERN Stack (MongoDB, Express, React, Node.js)**:

The MERN stack is well-suited for this project due to its flexibility, performance, and full-stack JavaScript nature. React will manage the front-end components for a responsive user interface, while Node.js and Express will handle backend logic. MongoDB will be used as the database due to its scalability and ease of integration with Node.js.

**Tailwind CSS**:

For fast, responsive, and modern UI design.

**REST APIs**:

To facilitate communication between the front-end and back-end.

**JWT (JSON Web Tokens)**:

For secure authentication.

2.2. Schema Design Considerations

In MongoDB, the schema will be structured to optimize data retrieval and storage. Key collections will include:

1. **userModel**: Storing customer and restaurant owner data.
2. **profilemodel**: creating a profile of a cutomer after logging the application.
3. **foodModel**: to store te data of food’s name, descriptins
4. **RestaurantsModel**: Storing restaurant details, menus, and related information.
5. **OrdersModel**: Tracking customer orders with status updates.
6. **DeliveryBoyModel: storing info about delivary boy with his availablity status.**
7. **ReviewsModel**: Storing ratings and customer feedback.

For example, the Orders collection will use references to the Users and Restaurants collections to minimize redundancy and optimize lookups. This schema design will also make it easier to scale the platform and manage data relationships efficiently. Similary there are other model also takes references from each other for maintaining maximum optimizaation.

## ****2.3 User Interactions****

**Customers**: Users will browse the restaurant listings and menus stored in MongoDB, place orders, and receive real-time updates. MongoDB’s flexible schema makes it easy to store varying menu items and special promotions.

**Restaurant Owners**: Restaurant managers will interact with the system to update their listings and manage orders. The MongoDB relationships between Orders and Restaurants will ensure that only relevant data is displayed to each restaurant.

**Admins**: Admins will manage users, restaurants, and orders through a dashboard that interacts directly with the MongoDB collections, enabling efficient content management and user administration.

# 3. Expected Challenges and Solutions

## ****3.1.**** Potential Challenges

**Scalability**: Handling increasing numbers of users, restaurants, and orders as the platform grows.

**Real-Time Updates**: Implementing real-time order tracking and delivery status.

**Data Consistency**: Ensuring that orders are accurately processed and tracked across different system components.

**Security**: Protecting user information, particularly payment data and personal details.

**Cross-Platform Performance**: Ensuring that the site performs well on all devices, including mobile and desktop.

## ****3.2.**** Proposed Solutions

**Database Sharding**: MongoDB’s sharding capabilities can be employed to scale the platform as it grows.

**WebSockets**: Implementing WebSockets for real-time communication, allowing users to track their orders live without needing to refresh the page.

**ACID Transactions**: For critical processes like order confirmation and payment, MongoDB’s multi-document ACID transactions can be used to ensure data consistency.

**Secure Payment and Data Handling**: Integrating HTTPS, JWT for user authentication, and the Stripe API for secure payments will mitigate security risks.

**Responsive Design and Caching**: Tailwind CSS will ensure the site is fully responsive, while caching mechanisms like Redis will optimize performance across devices.

# 4. Expected Outcomes and Benefits

## ****4.1. Impact on Target Audience****

**Convenience**: Ekhana will offer users a seamless way to order food from a variety of restaurants, saving them time and effort.

**Access to Local Eateries**: Smaller restaurants will benefit from increased exposure and the ability to offer delivery without needing their own infrastructure.

**Transparency**: Real-time order tracking will give users a sense of control and assurance, improving trust in the platform.

**Customer Feedback**: The review and rating system will help users make informed decisions and provide valuable feedback to restaurants.

## 4.2 Learning Objectives

**By completing this project, I aim to achieve the following:**

**Technical Skills**: Gain expertise in full-stack web development using the MERN stack, real-time features, and database optimization.

**User Experience (UX) Design**: Learn to design user-friendly interfaces and build intuitive, responsive applications.

**Backend Development**: Improve my ability to design efficient database schemas, handle complex data relationships, and implement secure, scalable REST APIs.

**Professional Skills**: Enhance problem-solving skills and develop a deeper understanding of how to approach complex project requirements, from technical solutions to user experience considerations.

# 5. Conclusion

This proposal outlines the detailed plan for the development of **Ekhana**, an online food delivery platform designed to offer users a seamless experience while supporting local restaurants. Through this project, I expect to tackle challenging technical issues and create a robust, scalable system that will benefit both users and restaurant owners.