International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Veggio : A MERN-Stack-Based Food Ordering Platform for Sustainable and Seasonal Eating

[1]Gunjan, [2]Vanshika Verma, [3]Vaishno Kumar Singh, [4]Shivam, [5]Sumit

[12]Department of Computer Science and Engineering (AI & ML)
[345]Raj Kumar Goel Institute of Technology , 5 km stone, Ghaziabad - 201003 projectveggio@gmail.com

ABSTRACT:

In the era of fast-paced lifestyles, online food ordering platforms have become an integral part of modern living. However, many platforms lack emphasis on sustainability and awareness of seasonal produce. This paper presents Veggio, a web-based food ordering platform developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). Veggio aims to bridge the gap between convenience and conscious consumption by recommending seasonal food items tailored to user locations. The application leverages a user-friendly interface, robust backend, and scalable database to ensure seamless functionality.

Keywords- MongoDB, Express.js, React.js, Node.js, backend,frontend, database.

INTRODUCTION

The rapid growth of e-commerce has extended into the food industry, leading to a proliferation of online food ordering platforms. These platforms have revolutionized how consumers interact with restaurants and grocery stores, offering convenience and efficiency. However, existing systems often lack emphasis on sustainability and fail to encourage users to make informed, eco-friendly choices. Seasonal eating, which aligns consumption with local agricultural cycles, is a crucial step toward sustainability but remains underrepresented in mainstream platforms.This paper introduces Veggio, a MERN-stack-based food ordering platform designed to address these gaps. Veggio not only facilitates efficient food ordering but also promotes sustainable practices by recommending seasonal and locally sourced food options. The platform's unique selling points include a responsive interface powered by React.js, a secure backend implemented using Node.js and Express.js, and a robust MongoDB database for scalable data management. It can either be stepped up or down as per the load and requirements and is ligthweight for both servers as storage efficiency and requires light computation power.

LITERATURE REVIEW

The growing reliance on online platforms for food delivery and grocery shopping has led to the development of several applica tions like Zomato, Swiggy, and BigBasket. These platforms rely on robust technical frameworks to handle complex operations, such as real time user interactions,secure payments gateways and inventory management.
Frontend Technologies: React.js and Angular are widely used for building dynamic and responsive user interfaces. React.js, in particular, stands out for its component-based architecture and ability to handle large-scale applications with ease.

Backend Architectures: Platforms like Swiggy use frameworks such as Node.js and Django to manage APIs, ensure scalability, and handle millions of transactions efficiently.

Database Management: Relational databases like MySQL and non-relational databases like MongoDB are popular choices. MongoDB, being a NoSQL database, is particularly well-suited for applications like Veggio due to its ability to handle unstructured data and scalability requirements.

Recommendation Engines: Recommendation systems employed in platforms like Netflix and Amazon are powered by machine learning algorithms. For seasonal and location-based recommendations, simpler approaches involving rule-based logic or weather APIs can suffice for a food ordering system like Veggio.

While these technologies form the backbone of current systems, existing food platforms often lack integration of sustainability features, such as dynamic recommendations based on seasonal availability and location. This gap provides an opportunity to create a platform that combines convenience with eco-consciousness, utilizing modern web technologies.

MOTIVATION AND OUR CONTRIBUTIONS

A - Motivation

- Inspired by noticing that most online food delivery platforms tend to overlook sustainability, our team developed Veggio to encourage eco-friendly choices without sacrificing convenience. We observed that existing systems rarely motivate users to select seasonal or locally-sourced foods, so we set out to fill this gap. Specifically, our motivation included the following factors:

The following factors motivated us:

- Promoting Sustainable Eating Habits: Seasonal produce offers benefits like reduced environmental impact, fresher ingredients, and support for local farmers. Yet most mainstream platforms do not highlight these options to users.
- Bridging the Technology Gap: Leading services (e.g., Zomato, Swiggy) excel in logistics but focus less on raising user awareness or offering eco-friendly features. Veggio was designed to incorporate engaging tools that foster environmental consciousness during the ordering process.
- Collaborative Teamwork: This project also provided an opportunity to apply and enhance our collective skills. By assigning specific roles—such as frontend design, backend development, database management, and user testing—we leveraged each member's strengths, making the development process both educational and efficient.
- By addressing these motivations, Veggio delivers a seamless and scalable user experience using modern web technologies, effectively filling the gaps in current food ordering platforms.

B - Our Contributions

- Our Veggio platform introduces several key contributions that merge sustainability with modern web technology, including:
- Development of a Seasonal Recommendation Engine : Veggio dynamically suggests menu items based on the current season and the user's location. This feature uses a lightweight rule-based logic integrated with weather APIs and seasonal datasets, encouraging users to choose locally available produce.
- Role-Based team Development: We structured the development effort so each team member could focus on a particular area. For example:
- Frontend Lead: Designed a responsive, intuitive interface using React.js for a smooth user experience.
- Backend Developer: Built scalable RESTful APIs and implemented user authentication using Node.js and Express.js.
- Database Administrator: Organized and managed MongoDB collections for users, orders, and seasonal food data, ensuring data integrity and efficient queries.
- Testing and Quality assurance: A designated team member conducted performance and usability tests under various conditions to ensure the platform runs smoothly and reliably.
- User and Admin Functionality: Veggio offers user-friendly features for customers (browsing, ordering, seasonal recommendations) and robust tools for admins (inventory management, seasonal data updates).
- Performance Optimization: We optimized both the frontend and backend to minimize page load times and improve responsiveness.
- Scalability and Deployment: The system is designed to be easily scalable. The architecture can accommodate future enhancements (such as machine-learning personalization.
- By leveraging the MERN stack and emphasizing teamwork, Veggio not only simplifies food ordering but also advocates for sustainable practices, setting it apart from traditional platforms.

V. METHODOLOGY

The methodology of Veggio involves the following steps:

High-Level Architecture

- We implemented Veggio with a modular architecture that cleanly separates the frontend, backend, and database layers. This design makes it easier to develop, maintain, and scale the system. The main components are:
- Frontend: Developed with React.js to create a dynamic, responsive interface. The frontend communicates with the backend via RESTful APIs.
- Backend: Built using Node.js with Express.js, this layer handles business logic and provides API endpoints for the application.

- Database: MongoDB is chosen for its scalability and flexibility in managing unstructured data (such as user profiles, orders, and seasonal information).

Frontend Design

- The user interface is implemented in React.js and includes features such as:
- User Dashboard: Displays personalized seasonal food recommendations, order history, and user settings.
- Responsive Design: Ensures compatibility across devices using modern CSS frameworks (e.g., Tailwind CSS).
- React Hooks: Usesj hooks (e.g., useState,

Integration

- The system integrates with external services and deployment tools, including:
- Third-Party APIs: Weather APIs provide data used by the recommendation engine.
- Payment Gateway: Integration with secure payment providers to handle transactions.
- Deployment: The application is containerized with Docker and managed with Kubernetes to ensure scalability and ease of deployment.

SYSTEM MODEL

The proposed system for Veggio, a MERN- stack-based food ordering website, is designed with a three-tier architecture comprising the frontend (React.js), backend (Node.js and Express.js), and database (MongoDB). This architecture ensures scalability, responsiveness, and efficient data handling for seamless user interaction.
Frontend (React.js)
useEffect) to manage application state and update content dynamically.

Backend Development

The Express.js server manages API routes and middleware. Key aspects include:
- Authentication and Authorization: Utilizes JSON Web Tokens (JWT) for secure user login and role-based access control (user vs. admin).
- Order Management: Provides RESTful endpoints for creating, updating, and retrieving orders.
- Recommendation Engine: Fetches seasonal food data via weather APIs or predefined seasonal datasets to power the recommendation engine.

Database Schema

The MongoDB database defines collections such as:
- Users: Stores user data (name, email, address, password hash).
- Orders: Tracks user orders with details like items, quantity, and timestamps.
- Products: Contains food item details, availability, and pricing.

The frontend provides a dynamic interface with reusable components for browsing products, viewing seasonal recommendations, and placing orders. State management is handled using React Context API or Redux, ensuring consistency across the application. Real-time data updates are enabled via API calls, allowing seamless interactions, such as live cart updates and user activity tracking. Error handling on the client- side ensures a smooth experience with user- friendly messages for failed operations.

Backend (Node.js and Express.js)

The backend serves as the core application layer, managing business logic and exposing RESTful APIs for critical functionalities like user authentication, order processing, and product recommendations.It employs asynchronous operations to optimize response times for high- traffic scenarios and provides centralized logging to monitor and debug backend activities effectively.

Database (MongoDB)

MongoDB stores data in collections for Users, Orders, Products, and Seasons. Flexible document structures and indexing enable fast queries and scalability, supporting dynamic seasonal recommendations and real-time order tracking.

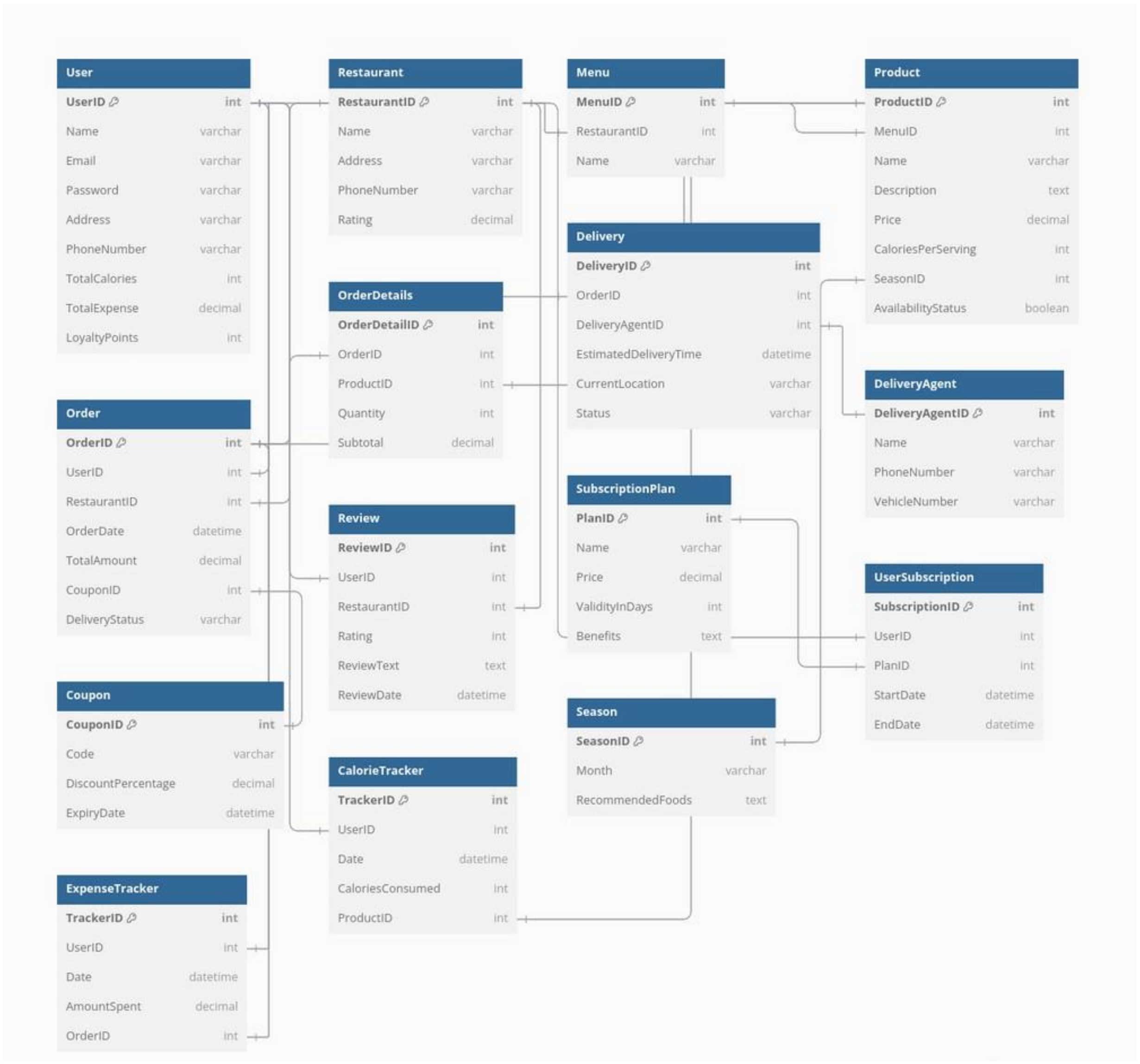

Fig .01

The ER diagram for Veggio in Fig.01 illustrates the relationships between core entities in the food ordering system. It consi sts of the following key components:

- User: Stores customer information such as name, email, and address. It tracks user-specific data like loyalty points, total expenses, and calorie consumption.
- Restaurant: Represents restaurants offering products, including their address, contact information, and average rating.
- Menu and Products: Menus are linked to restaurants, and products are associated with menus, storing details like price, calories, and seasonal availability.
- Order and Order Details: Captures user orders, linking them to restaurants and products, along with details like quantity and applied coupons.
- Delivery: Tracks delivery status and is linked to delivery agents, including live location and estimated delivery time.
- Review: Allows users to rate and review restaurants.
- Subscription Plans: Enables users to subscribe to plans with benefits like discounts or free delivery.
- Calorie and Expense Tracker: Tracks daily user expenses and calorie intake, linking orders and products.
- Coupons: Stores promotional codes with discount details and expiry dates.
- Seasons: Associates seasonal recommendations with products for sustainability.
- This ER diagram showcases a modular and relational structure to efficiently manage user interactions, product data, and system scalability.

CONCLUSIONS AND FUTURE SCOPE

A - Conclusion

- Veggio demonstrates how combining modern web technologies with sustainability can yield a powerful food ordering platform. Using a modular MERN-stack architecture, Veggio provides an efficient, user-friendly, and scalable solution that not only simplifies online ordering but also encourages eco-friendly eating habits.

B - Future Scope

- Veggio's unique features (such as seasonal recommendations, coupon and loyalty systems, and calorie/expense tracking) already distinguish it from traditional platforms. Future enhancements could include:
- Personalized Recommendations: Integrating machine learning algorithms to tailor food suggestions to individual user preferences and habits.
- Mobile and Multi-language Support: Expanding the platform to native mobile apps and additional languages to reach a broader audience.
- Blockchain for Transparency: Exploring blockchain solutions to increase transparency in the food supply chain and delivery process, further promoting trust and sustainability.

REFERENCES:

1. MongoDB Documentation: MongoDB, Inc."MongoDB Manual." Accessed at https://www.mongodb.com/docs/.
2. This reference was used to design the database schema and optimize queries for scalability and performance.
3. React.js Documentation: Meta Platforms, Inc. "React – A JavaScript library for building user interfaces." Accessedat https://reactjs.org/docs/getting-started.html.
4. Node.js Documentation: OpenJS Foundation. "Node.js – JavaScript runtime." Accessed at https://nodejs.org/en/docs/.
5. Used to develop the server-side logic and asynchronous operations for the backend.
6. Express.js Documentation: OpenJS Foundation. "Express – Fast, unopinionated, minimalist web framework for Node.js." Accessed at https://expressjs.com/.
7. Served as a reference for designing RESTful API endpoints and implementing middleware for authentication and error handling.
8. Zomato and Swiggy Platforms: Zomato Media Pvt Ltd and Bundl Technologies Pvt Ltd. Used as benchmarks to analyze gaps in existing food delivery platforms and to identify unique features for Veggio.
9. Sustainability in Food Systems: Evans, D. "The Importance of Seasonal Food Consumption." Journal of Sustainable Food Practices, 2021.
10. Inspired the integration of seasonal recommendations into Veggio to promote environmentally conscious eating habits.
11. Web Performance Optimization: Google Developers. "Lighthouse – Web Performance Tools." Accessed at https://developers.google.com/web/tools/lighthou se.
12. Assisted in optimizing page load speeds and frontend performance metrics.
13. API Testing Tools: Postman, Inc. "Postman API Platform." Accessed at https://www.postman.com/.
14. Used extensively for testing RESTful APIs to ensure reliability and performance.
15. Machine Learning in Food Recommendations: Sharma, A., and Gupta, R. "Enhancing Food Delivery Systems with AI." Proceedings of the International Conference on Data Science, 2022. Explored as potential future work for personalized food recommendations in Veggio.
16. Design Principles: Nielsen, J. "10 Usability Heuristics for User Interface Design." Nielsen Norman Group, 2020.
17. Provided guidance on creating an intuitive and user-friendly interface for Veggio.
18. OpenWeather API Documentation: OpenWeather Ltd. "Current Weather and Forecast API." Accessed at https://openweathermap.org/api.
19. Used to fetch location-based weather data for seasonal recommendations.