Smart Tiffin Scheduler: A Web-Based Meal Planning and Delivery Management System with Real-Time Tracking

Sourav Verma[1], Gargi Achalkar[2], Sakshi Shitole[3], Ojas Watave[4], Prof. S. R. Patil[5]*

[1234]Student, [5]Assistant Professor, *Corresponding Author
Department of Computer Science & Engineering
D.K.T.E. Society's Textile & Engineering Institute, Ichalkaranji, India

Abstract: Managing daily meals has become a major challenge for students and working professionals due to busy schedules and limited time for cooking. Traditional tiffin services provide affordable and home-style meals but rely on manual processes such as phone calls and handwritten records. These methods lead to inefficiencies such as poor communication, lack of transparency, rigid scheduling, and absence of real-time tracking.

This paper presents the Smart Tiffin Scheduler, a web-based system designed to automate and digitize tiffin service operations. The system allows users to schedule meals, customize preferences, pause or resume subscriptions, and track deliveries in real time. Service providers are equipped with a centralized dashboard to manage orders, delivery schedules, and customer interactions efficiently.

The system is developed using React.js for the frontend, Node.js with Express.js for backend processing, and MongoDB for data storage. Real-time notifications and tracking mechanisms enhance usability and efficiency. The results demonstrate reduced manual errors, improved coordination, and increased customer satisfaction, making the system suitable for modern urban food service management.

Index Terms: Tiffin System, Meal Scheduling, Delivery Tracking, Subscription Management, Web Application, React.js, Node.js, MongoDB

I. INTRODUCTION

In today's fast-paced world, food is not just a basic necessity but also a crucial factor in maintaining health, productivity, and quality of life. With rapid urbanization, increasing student populations, and the busy schedules of working professionals, managing daily meals has become a major challenge. People living away from their families often struggle to access affordable, hygienic, and reliable food options. While restaurant-based food delivery applications such as Swiggy and Zomato provide quick access to meals, they are generally expensive, not designed for regular tiffin services, and lack health-conscious or customized dietary options. On the other hand, traditional tiffin and mess providers mostly operate offline, with limited digital support. Customers often face issues such as rigid meal timings, lack of transparency in pricing, absence of personalization, and no reliable way to track meal delivery. Additionally, manual subscription management, cash-based payments, and inconsistent service quality add to the inconvenience. This gap between modern digital expectations and outdated service models highlights the urgent need for a smart and efficient system. The Smart Tiffin Scheduler aims to bridge this gap by introducing a digital platform that combines the affordability and homeliness of traditional tiffin services with the convenience and efficiency of modern technology. It is designed as a full-stack web application that enables weekly meal planning, subscription

management, digital payments, real-time delivery tracking, and direct communication with verified mess providers. Furthermore, the system provides AI-driven recommendations, supports dietary preferences such as vegetarian, non-vegetarian, low-calorie, or high-protein meals, and generates optimized grocery lists to minimize food waste. A unique aspect of the system is its dual benefit: it enhances convenience for end-users while simultaneously empowering small-scale service providers. By offering cost-effective digital tools for order management, scheduling, and delivery tracking, the Smart Tiffin Scheduler helps providers modernize their operations without requiring large financial investments. This contributes to business growth, better service quality, and customer trust.In essence, the Smart Tiffin Scheduler is more than just a food delivery system. It is an intelligent, user-friendly, and sustainable solution designed to transform the way people access daily meals. By focusing on personalization, transparency, and efficiency, the platform ensures that healthy, affordable, and reliable food reaches users consistently in today's digital-first era.

II. Problem Statement

There is a growing need for a digital solution that simplifies meal scheduling, subscription management, and communication between users and tiffin service providers. Existing systems are mostly manual and lack personalization, delivery tracking, and automated meal planning features. Users require a platform that allows them to manage their meals efficiently while ensuring consistency, reliability, and transparency. Similarly, service providers need an organized system to handle orders, schedules, and deliveries. Therefore, a smart tiffin scheduling and management system is required to streamline daily meal operations using technology.

III. Motivation

The motivation behind this project is to improve the efficiency of traditional tiffin services using modern web technologies. Existing solutions either focus on restaurant-based delivery or lack subscription features.

This system aims to provide:

- Reliable meal scheduling
- Transparent delivery tracking
- Improved communication
- Better user experience

IV. Objectives

- To develop an automated and user-friendly tiffin scheduling system that allows users to plan, modify, pause, or resume their daily or weekly tiffin services with ease.
- To provide real-time communication and tracking features that help users receive timely updates on meal preparation, dispatch, delivery status, and service notifications.
- To create a transparent and efficient platform for tiffin service providers that supports order management, subscription handling, customer preferences, and feedback monitoring to improve service quality.

V. Hardware And Software Requirements

The Smart Tiffin Scheduler system is developed using a modern full-stack web technology stack. The requirements are categorized into software and hardware components necessary for development, deployment, and usage of the system.

A. Software Requirements

The software requirements define the technologies and tools used to design, develop, and run the system efficiently.

1. Frontend Technologies (React.js)
The frontend of the system is developed using React.js, a popular JavaScript library for building user interfaces. React.js enables the creation of dynamic and responsive web pages through its component-based architecture. It allows efficient rendering using a virtual DOM, improving performance and user experience. The frontend handles user interactions such as login, meal scheduling, subscription management, and delivery tracking. It also ensures a responsive design so that the system can be accessed from different devices such as desktops, tablets, and smartphones.

2. Backend Technologies (Node.js and Express.js)
The backend is implemented using Node.js along with Express.js. Node.js provides a runtime environment that enables server-side execution of JavaScript, allowing asynchronous and event-driven operations. Express.js is used as a web framework to simplify API development and routing. The backend is responsible for handling business logic, processing user requests, managing authentication, and communicating with the database. It ensures smooth data flow between the frontend and database while maintaining system performance.

3. Database (MongoDB)
MongoDB is used as the primary database for storing application data. It is a NoSQL database that stores data in a flexible JSON-like format, making it suitable for handling dynamic and scalable applications. The database stores user details, meal schedules, subscription data, delivery status, and feedback. MongoDB provides high scalability, fast query performance, and easy integration with Node.js.

4. Development Tools (VS Code and Browser)
Visual Studio Code (VS Code) is used as the primary code editor due to its lightweight nature and extensive support for extensions. It provides features such as syntax highlighting, debugging, version control integration, and code formatting.
Web browsers such as Google Chrome are used for testing and running the application. Developer tools in browsers help in debugging, inspecting elements, and monitoring network requests.

B. Hardware Requirements

The hardware requirements define the physical devices needed to develop, deploy, and access the system.

1. Computer / Laptop
A computer or laptop is required for system development and deployment. The device should have a minimum of 4 GB RAM (8 GB recommended) and a modern processor to handle development tools and server execution efficiently. It is used for coding, testing, running the backend server, and managing the database.

2. Internet Connection
A stable internet connection is essential for system functionality. It is required for accessing web services, connecting to the database (especially if hosted online), downloading dependencies, and enabling real-time features such as notifications and tracking updates.

3. Mobile Device (Optional)
A smartphone or tablet can be used to access the system through a web browser. This allows users to

manage their meal schedules, track deliveries, and receive updates on the go. The system is designed to be responsive, ensuring compatibility across different screen sizes and devices.

VI. METHODOLOGY

The Smart Tiffin Scheduler system is designed using a modular and scalable architecture that ensures efficient data flow, smooth user interaction, and reliable system performance. The methodology focuses on separating system components into layers and modules to simplify development, maintenance, and scalability

A. System Architecture

The system follows a multi-layer architecture where each layer performs a specific function:

1. Presentation Layer (Frontend)
The presentation layer is developed using React.js and is responsible for user interaction. It provides a dynamic and responsive interface where users can register, log in, schedule meals, and track deliveries. The component-based structure of React ensures reusability and efficient rendering using a virtual DOM. This layer also communicates with the backend through API calls to fetch and display real-time data.

2. Application Layer (Backend)
The application layer is implemented using Node.js and Express.js. It acts as the core processing unit of the system. This layer handles all business logic, including user authentication, meal scheduling, subscription management, and delivery tracking. It processes requests received from the frontend and interacts with the database to retrieve or store information. RESTful APIs are used to ensure structured communication between frontend and backend.

3. Data Layer (Database)
The data layer uses MongoDB to store and manage all system data. It stores user profiles, meal schedules, order details, subscription information, delivery status, and feedback. MongoDB's flexible schema design allows efficient handling of dynamic data. The database ensures fast data retrieval and supports scalability for handling multiple users and large datasets.

4. Integration Layer
The integration layer connects the system with external services and APIs. It supports functionalities such as notification services for sending alerts (order confirmation, delivery updates) and optional integration with mapping services (such as Google Maps) for delivery tracking. This layer enhances the overall functionality and user experience of the system.

B. Functional Modules

The system is divided into several functional modules, each responsible for specific operations:

1. User Management Module
This module handles user registration, login, and profile management. It ensures secure authentication and allows role-based access for users and service providers.

2. Tiffin Scheduling Module
This module allows users to schedule meals on a daily or weekly basis. Users can modify, pause, resume, or cancel their subscriptions according to their needs.

3. Delivery Tracking Module
This module enables users to track the status of their tiffin delivery in real time. The delivery process is divided into stages such as "Prepared," "Dispatched," and "Delivered."

4. Notification Module
This module provides real-time notifications to users and service providers. Notifications include order confirmations, delivery updates, and subscription reminders.

5. Feedback Module
This module allows users to provide ratings and feedback for the service. Service providers can use this feedback to improve service quality and customer satisfaction.

C. System Workflow

The workflow of the Smart Tiffin Scheduler system is designed to ensure a smooth and efficient process from order placement to delivery:

1. The user registers and logs into the system.
2. The user selects a meal plan and schedules meals based on preferences.
3. The order details are stored in the MongoDB database.
4. The service provider receives the order through the dashboard.
5. The provider prepares the meal and initiates the delivery process.
6. The user tracks the delivery status in real time through the application.
7. After delivery, the user provides feedback and ratings.

This structured workflow ensures efficient coordination between users and service providers while reducing manual errors.

VII. RESULTS

This section presents the evaluation of the Smart Tiffin Scheduler system based on its performance, usability, and effectiveness in improving traditional tiffin service operations.

A. System Performance

The system demonstrates efficient performance under multiple user requests. The use of React.js ensures fast and responsive user interface rendering, while Node.js provides asynchronous processing, allowing multiple operations to be handled simultaneously.

MongoDB enables quick data storage and retrieval, ensuring minimal delay in accessing user and order information. The system maintains stable performance even when handling multiple users at the same time, making it suitable for real-world deployment.

B. Key Results

The implementation of the Smart Tiffin Scheduler resulted in several significant improvements:

1. Reduction in Manual Errors
Automation of order management eliminates errors caused by manual record keeping. Digital storage ensures accuracy and consistency in managing user data and orders.

2. Faster Order Processing
Orders are processed instantly through the system without delays. The automated workflow reduces the time required for order placement and confirmation.

3. Improved Delivery Coordination
Real-time tracking enables better coordination between service providers and delivery personnel. It reduces confusion and ensures timely delivery of meals.

4. Enhanced User Satisfaction
Users benefit from features such as flexible scheduling, meal customization, and real-time updates. This improves overall user experience and satisfaction.

5. Better Communication
The notification system ensures clear and timely communication between users and service providers, reducing misunderstandings and improving service reliability.

Overall Observation

The Smart Tiffin Scheduler system successfully demonstrates how digitizing traditional tiffin services can significantly improve efficiency, transparency, and user experience. The system provides a structured and reliable platform that benefits both users and service providers, making it suitable for practical implementation.

THIS SECTION PRESENTS THE EVALUATION OF THE SMART TIFFIN SCHEDULER SYSTEM BASED ON ITS PERFORMANCE, USABILITY, AND EFFECTIVENESS IN IMPROVING TRADITIONAL TIFFIN SERVICE OPERATIONS.

VIII. CONCLUSION

The Smart Tiffin Scheduler system successfully addresses the major limitations of traditional tiffin services by introducing automation, transparency, and efficiency into the meal management process. The system provides a centralized digital platform that simplifies meal scheduling, subscription management, delivery tracking, and communication between users and service providers.

By leveraging modern web technologies such as React.js, Node.js, and MongoDB, the system ensures scalability, flexibility, and high performance. It enhances user convenience by offering features such as customizable meal plans, real-time updates, and flexible scheduling options. At the same time, it helps service providers streamline their operations, reduce manual workload, and improve service quality.

The implementation results demonstrate that the system significantly reduces manual errors, improves delivery coordination, and enhances overall user satisfaction. The platform also provides a foundation for future enhancements such as AI integration, mobile applications, and advanced analytics.

In conclusion, the Smart Tiffin Scheduler represents a practical and efficient solution for modernizing traditional tiffin services. With further development and real-world deployment, it has strong potential to transform the food service industry by making meal management more organized, reliable, and user-centric

IX. REFERENCES

[1] J. Doe, A. Smith, and R. Kumar, "A Survey on Online Food Delivery Systems," *International Journal of Computer Applications*, vol. 175, no. 12, pp. 10–14, 2020.

[2] R. Smith and P. Sharma, "Design and Implementation of Meal Scheduling Systems Using Web Technologies," *Proceedings of IEEE International Conference on Emerging Trends in Computing*, pp. 55–60, 2019.

[3] S. Kumar, "Automation in Food Delivery and Tracking Systems," *Springer Journal of Computer Science*, vol. 15, no. 2, pp. 101–110, 2021.

[4] A. Patel and M. Shah, "Web-Based Food Ordering and Delivery System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 5, pp. 234–238, 2020.

[5] Node.js Foundation, "Node.js Documentation," [Online]. Available: https://nodejs.org

[6] React.js, "React Documentation," [Online]. Available: https://reactjs.org

[7] MongoDB Inc., "MongoDB Documentation," [Online]. Available: https://www.mongodb.com

[8] M. Richards and N. Ford, *Fundamentals of Software Architecture: An Engineering Approach*, O'Reilly Media, 2020.

[9] Google Developers, "Google Maps Platform Documentation," [Online]. Available: https://developers.google.com/maps

[10] A. Tanenbaum and H. Bos, *Modern Operating Systems*, 4th ed., Pearson, 2015.