**Industrial Internship Report**

## Tatvasoft

***Submitted by***

# Vivek Raiyani

### 12102080601146

***In partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

***in***

**Information Technology A.D.Patel Institute of Technology**

|  |  |
| --- | --- |
|  | CVM University, Vallabh Vidyanagar Recruitment 2020 |

# Charutar Vidya Mandal University

**April, 2024**

|  |  |
| --- | --- |
|  | CVM University, Vallabh Vidyanagar Recruitment 2020 |

# A.D.Patel Institute of Technology

CERTIFICATE

This is to certify that **Raiyani Vivek (12102080601146)** has submitted the Industrial Internship report based on internship undergone at **TatvaSoft** for a period of **16** weeks from **20/1/2025** to **9/5/2025** in partial fulfillment for the degree of Bachelor of Engineering in **Information Technology A.D.Patel Institute of Technology** at The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar during the academic year 2023 – 24.

|  |  |
| --- | --- |
| Prof. Anu Chauhan | Dr. Narendrasinh C Chauhan |
| signature | Head of the Department, IT |

****

# 

# A.D.Patel Institute of Technology New Vallabh Vidhyanagar

**DECLARATION**

I, Raiyani Vivek 12102080601146, hereby declare that the Industrial Internship report submitted in partial fulfillment for the degree of Bachelor of Engineering in Information TechnologyA.D.Patel Institute of Technology, The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar, is a bonafide record of work carried out by me at Tatvasoft under the supervision of Anu Chauhan and that no part of this report has been directly copied from any students’ reports or taken from any other source, without providing due reference.

Name of the Student Sign of Student

Raiyani Vivek

# List of Figures

Fig 1 loginScreen 19

Fig 2 OrderList 20

Fig 3 Menu Page 21

Fig 4 Invoice Pdf 22

Fig 5 Dashboard Page 23

Fig 6 Table Module Order App 24

Fig 7 Table Module Order App 25

Fig 8 Menu Module Order App 25

Fig 9 Waiting List Order App 26

Fig 10 KOT Order App 27

# Table of Contents

Acknowledgement i

List of Figures vi

List of Abbreviations viii

Table of Contents vii

##### Chapter 1 Company Overview 9

1.1 Introduction about the Company 9

1.2 Services Offered By Company 10

1.3 Sequence of Operation 10

##### Chapter 2 Introduction to Project 11

2.1 Project Specification 11

2.2 Platform Specification 12

2.3 Objective 13

2.4 Scope 13

##### Chapter 3 System Analysis 15

3.1 Current System 15

3.2 Weakness and Problem 15

3.3 System Feasibility 16

##### Chapter 4 System Design 17

4.1 System Design and Architecture 17

4.2 Database Scheme and Flows 17

4.3 Security and Authentication 18

##### Chapter 5 Implementation 19

3.1 Project Flow 19

3.2 System Requirements 28

3.3 Implementation of Project 28

##### Chapter 6 Conclusion 31

6.1 Summary 31

##### References 32

# 

# LIST OFABBREVIATIONS

|  |  |
| --- | --- |
| Docs | Documents |
| IDE | Integrated Development Environment |
| SQL | Structured Query Language |
| JS | JavaScript |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheet |
| CSHTML | C Sharp Hyper Text Markup Language |
| CS | C Sharp |
| DBA | Database Administrator |
| DBMS | Database Management System |
| HR | Human resources |
| QA | Quality Assurance |
| SMS | Short Message Service |
| UI | User Interface |
| UX | User Experience |

# Chapter 1

## Overview of the Company

### TatvaSoft

TatvaSoft was Established in the year of 2021 and is a globally recognized IT solutions provider offering software development services across a wide range of industries. With its headquarters in India and offices in multiple countries, TatvaSoft specializes in custom software development, including enterprise solutions, web development, mobile apps, and cloud services.

TatvaSoft’s mission is to deliver cutting-edge technology solutions that meet client requirements while ensuring quality and scalability. Over the years, the company has earned a reputation for delivering high-quality IT services to both SMEs (small and medium-sized enterprises) and large enterprises.

#### BRIEF INTRODUCTION ABOUT THE COMPANY

TatvaSoft is a CMMi Level 3 & Microsoft Gold partner Software Development Company offering software development services on diverse technology platforms like Microsoft, Angular, React, NodeJS, Java, PHP, SharePoint, Open Source, BI, & Mobile.

With rich and varied experience of 18 years in software development and stringent quality standards, TatvaSoft offer utmost qualitative, on-time and cost-effective software solutions. They serve clientele across the industries and globe with offices in US, Canada, UK, Australia, and Development Center in India with a workforce of 810 IT professionals.

#### SERVICES OFFERED BY COMAPNY:

* Custom Software Development
* Product Development
* Mobile Apps Development
* Web Development
* Software Testing / QA
* Enterprise Solutions
* Cloud Services
* Dedicated Development Center
* Product Maintenance
* Software Outsourcing
* UI / UX Design

**SEQUENCE OF OPERATORS**

* **Requirement Gathering & Analysis:** Our research team gathers your requirements, analyzes them, and defines the features to be developed. They also estimate the project's cost and timeline.
* **Development & Team Allocation:** We leverage expert teams like ASP.NET MVC for backend and a modern UI framework for frontend, ensuring a tailored approach for each project.
* **Testing & Quality Assurance:** Once development is complete, our testing team meticulously tests the software to ensure it meets all requirements and functions flawlessly.

# Chapter 2

## Introduction to internship/Project

### PizzaShop

The **Pizza Shop Management System** is a comprehensive, web-based application designed to optimize and automate critical restaurant management processes. This system integrates user management, table/section management, dynamic menu customization, role-based access control (RBAC), tax and fee computation, and real-time order processing into a centralized platform.

A key feature of this solution is its multi-role dashboard, which provides specialized interfaces for different users, including the **Super Admin, Chef, and Account Manager**, ensuring efficient operational oversight and informed decision-making. The project leverages a modular architecture to improve system scalability, maintainability, and extensibility, making it a robust solution for restaurant operations.

#### 2.1 Project Summary

The **Pizza Shop Management System** is designed to enhance both front-end and back-end operations by integrating multiple interconnected modules, streamlining daily restaurant workflows, and minimizing human intervention to reduce errors.

The system offers:

* Role-based user access control to enforce security and operational restrictions.
* A dynamic and scalable order processing mechanism that optimizes kitchen workflows.
* Comprehensive analytics and reporting dashboards to provide real-time business insights.

The project's objective is to digitize and automate restaurant operations while maintaining an intuitive, responsive user interface for seamless interaction across different roles.

#### 2.2 Platform Specification

The **Pizza Shop Management System** is engineered as a full-stack, web-based platform that can efficiently handle user interactions and data processing in real-time and a strong emphasis on **performance, security, and scalability**. The system is built using a **Model-View-Controller( MVC ) architecture**, ensuring maintainability.

##### 2.2.1 Frontend Technologies

The frontend is implemented using modern web technologies to provide a highly interactive, responsive, and user-friendly interface:

* **HTML5** – Used for structuring web pages and ensuring semantic, accessible content.
* **CSS3** – Implements adaptive styling to enhance the visual appeal and responsiveness of the UI.
* **JavaScript (ES6+)** – Facilitates dynamic interactivity and enhances client-side functionality.
* **Bootstrap** – A responsive CSS framework for consistent UI design across devices.

##### 2.2.2 Backend Technologies

The backend is built on **ASP.NET Core MVC**, providing high-performance, strong security mechanisms, and efficient request handling:

* **ASP.NET Core MVC** – Implements the Model-View-Controller (MVC) pattern, ensuring modularity, reusability, and separation of concerns.
* **C#** – A strongly-typed, object-oriented programming language used for business logic implementation, security enforcement, and data manipulation.
* **Entity Framework Core** – An ORM (Object-Relational Mapping) framework that simplifies database operations and improves query performance.

##### 2.2.3 Database Management System:

The system uses **PostgreSQL** with **PGADMIN** as the primary database management system due to its **scalability, performance**:

* **PostgreSQL** – A powerful, open-source relational database known for its **high availability, data integrity, and support for complex queries**.

This technology stack enables the **Pizza Shop Management System** to efficiently manage real-time user interactions, data processing, and business operations, ensuring **reliability, scalability, and security**

#### 2.3 Objectives

The primary objective of the Pizza Shop Management System is to streamline and optimize daily restaurant operations by leveraging modern technology to minimize human error, improve efficiency, and enhance decision-making.

To achieve this, the platform is designed as a centralized system that integrates various operational modules, enabling restaurant management to automate repetitive tasks, monitor real-time data, and maintain seamless workflow execution.

Key Objectives:

* Operational Efficiency – Reduce manual workload by automating core processes such as order management, user roles, menu updates, and invoice generation.
* Error Reduction – Implement structured workflows and real-time validation to minimize human errors in order processing, billing, and inventory tracking.
* Real-Time Monitoring – Enable restaurant staff to access live operational data, ensuring swift decision-making for order tracking, inventory management, and financial oversight.
* Role-Based Access Control (RBAC) – Restrict system functionalities based on user roles (Super Admin, Chef, Account Manager, etc.) to ensure security and efficiency.
* Scalability & Maintainability – Design the system with a modular and scalable architecture, allowing future expansions such as integration with third-party services (payment gateways, analytics tools, etc.).

By integrating these features, Pizza Shop aims to enhance restaurant management capabilities, reduce operational bottlenecks, and ensure seamless day-to-day business execution.

#### 2.4 Scope

The Pizza Shop Management System is developed exclusively for restaurant operations, focusing on the internal management aspects without incorporating customer-facing functionalities. The platform is not intended for customers to place orders, browse menus, or make reservations; instead, it serves as a back-end solution for restaurant staff and administrators.

Functional Scope:

The application is tailored to restaurant-side operations, covering:

* Reservation Management – Efficiently handle table bookings and section allocations.
* Order Processing – Streamline the ordering workflow from placement to kitchen preparation and billing.
* Menu & Inventory Management – Allow real-time updates to food items, pricing, availability, and stock levels.
* Invoice & Tax Handling – Automate billing, tax calculations, and financial record-keeping.
* User & Role Management – Assign and regulate access for different users, ensuring security and task-specific control

#### 2.5 Exclusions

* No Customer Interface – The system does not support direct customer interactions such as online ordering, feedback submissions, or mobile app-based reservations.
* No Third-Party Food Delivery Integration – The platform does not integrate with food delivery services like UberEats, Zomato, or Swiggy at this stage.
* No Customer Loyalty or Marketing Features – The application does not include loyalty programs, promotional campaigns, or customer engagement modules.

By focusing entirely on restaurant-side operations, the Pizza Shop Management System ensures a highly optimized and dedicated platform for restaurant administrators, chefs, and managers, ultimately improving overall business performance.

# Chapter3

## System Analysis

### 3.1 Current System

In traditional restaurant management, most processes are handled manually or through **basic point-of-sale (POS) systems** with limited automation. Many small to mid-sized restaurants rely on **paper-based order tracking, manual inventory management, and fragmented financial reporting**, leading to inefficiencies and potential errors.

**Common Features of Current Systems:**

* **Manual Order Processing** – Orders are written on paper or input into basic POS systems without real-time tracking.
* **Limited Inventory Management** – Stock levels are often checked manually, leading to inaccurate forecasting and overstocking/understocking issues.
* **Decentralized User Access** – Different departments (kitchen, front-of-house, finance) operate separately, with little synchronization.
* **Minimal Data Utilization** – Restaurants lack data-driven insights for optimizing menus, pricing, and waste reduction.

While some restaurants have started adopting **cloud-based POS and inventory systems**, most traditional setups lack the **scalability, automation, and advanced analytics** needed for modern restaurant operations.

### 3.2 Problems and Weaknesses

Despite existing restaurant management solutions, several limitations hinder operational efficiency:

**Operational Challenges:**

* **High Dependency on Manual Processes** – Human errors in order taking, billing, and inventory tracking lead to revenue loss.
* **Inconsistent Order Tracking** – No centralized dashboard to track real-time order statuses, leading to miscommunication between staff.
* **Delayed Inventory Updates** – Without automation, stock shortages or wastage are detected too late, affecting business profitability.

**Technological Weaknesses:**

* **Lack of Integration with Delivery Platforms** – Many traditional systems do not integrate with **Uber Eats, Zomato, or Swiggy**, causing inefficiencies in online order processing.
* **Security Risks** – Manual access control systems make sensitive financial and operational data vulnerable to unauthorized access.
* **Scalability Issues** – On-premise systems struggle to support **multi-location restaurant chains**, limiting business expansion.

### 3.3 System Feasibility

Before implementing the **Pizza Shop Management System**, a **feasibility study** was conducted to assess its viability in terms of **technical, operational, and economic factors**.

**1. Technical Feasibility**

The system is built on **modern, scalable technologies**, ensuring robust performance:

* **Cloud-based architecture** for real-time data synchronization.
* **AI-driven analytics** for data-based decision-making.
* **Integration with external platforms** for seamless order processing.

**2. Operational Feasibility**

The system is designed to fit into existing restaurant workflows:

* **Minimal learning curve** due to a user-friendly interface.
* **Role-based access control** ensures secure operations.
* **Automated inventory and order management** reduces workload and errors.

**3. Economic Feasibility**

The system provides a cost-effective solution by:

* Reducing **manual labor costs** through automation.
* **Minimizing food wastage** with predictive inventory tracking.
* Increasing revenue through **data-driven menu optimization**.

By addressing the **limitations of existing restaurant management systems** and offering an **automated, AI-enhanced, and scalable solution**, the **Pizza Shop Management System** proves to be **technically, operationally, and economically feasible** for modern restaurant businesses.

# Chapter 4

## System Design

### 4.1 System Design & Architecture

The Pizza Shop Management System follows a **three-tier architecture**, ensuring scalability, security, and maintainability. The system is structured into the following layers:

1. **Presentation Layer:**  This layer is responsible for the user interface and interaction. It is built using **HTML, CSS, JavaScript, and Bootstrap**, ensuring a responsive and user-friendly design.
2. **Business Logic Layer:** The core functionalities and business rules are implemented in **ASP.NET Core 7 MVC**, which provides a robust and scalable backend framework.
3. **Data Layer:** The system uses **PostgreSQL** as the primary database for structured data storage, ensuring high performance and data integrity.

This architecture enables seamless communication between the frontend, backend, and database, supporting efficient restaurant operations and management.

### 4.2 Database Schema & Data Flow

The database schema is designed to efficiently manage restaurant operations, ensuring data consistency and integrity. The key tables include:

* Users Table: Stores user credentials, roles, and access permissions, ensuring secure authentication and role-based access.
* Orders Table: Tracks order status, payment details, and delivery status, ensuring real-time monitoring of restaurant operations.
* Menu Table: Contains food items, categories, and modifiers, allowing dynamic menu updates and modifications.
* Inventory Table: Monitors stock levels, supplier data, and usage history, ensuring proper inventory management and reducing waste.
* Payments Table: Manages transactions, refunds, and invoice records, supporting secure and reliable financial transactions.

The data flow ensures that every operation is logged and updated in real-time, providing accurate reporting and analytics for management.

### 4.3 Security & Authentication

Security is a key component of the system, ensuring data protection and secure user access. The following measures are implemented:

* JWT-based authentication: Ensures secure user sessions and prevents unauthorized access.
* Role-Based Access Control (RBAC): Restricts functionalities based on user roles such as Admin, Chef, and Manager, improving security and access management.
* AES encryption: Encrypts sensitive customer and financial data, ensuring data confidentiality.
* Multi-Factor Authentication (MFA): Adds an additional layer of security for user logins, preventing unauthorized access.
* Audit logs: Maintain records of modifications and activities, enabling monitoring and fraud detection.

These security measures collectively enhance the reliability and integrity of the system, protecting both user data and restaurant operations.

# Chapter 5

## Implementation

### 5.1 Project Flow

The **Pizza Shop Management System** follows a structured flow to ensure seamless restaurant operations. The process begins with user authentication and role-based access control, followed by order processing, inventory management, and financial transactions. This systematic approach helps in improving efficiency and reducing manual errors.

* **User Authentication & Authorization:**

Users log in using **JWT-based authentication**, ensuring secure session management.

Role-Based Access Control (RBAC) assigns different access levels (Admin, Chef, Manager, etc.), restricting unauthorized access to sensitive functionalities.

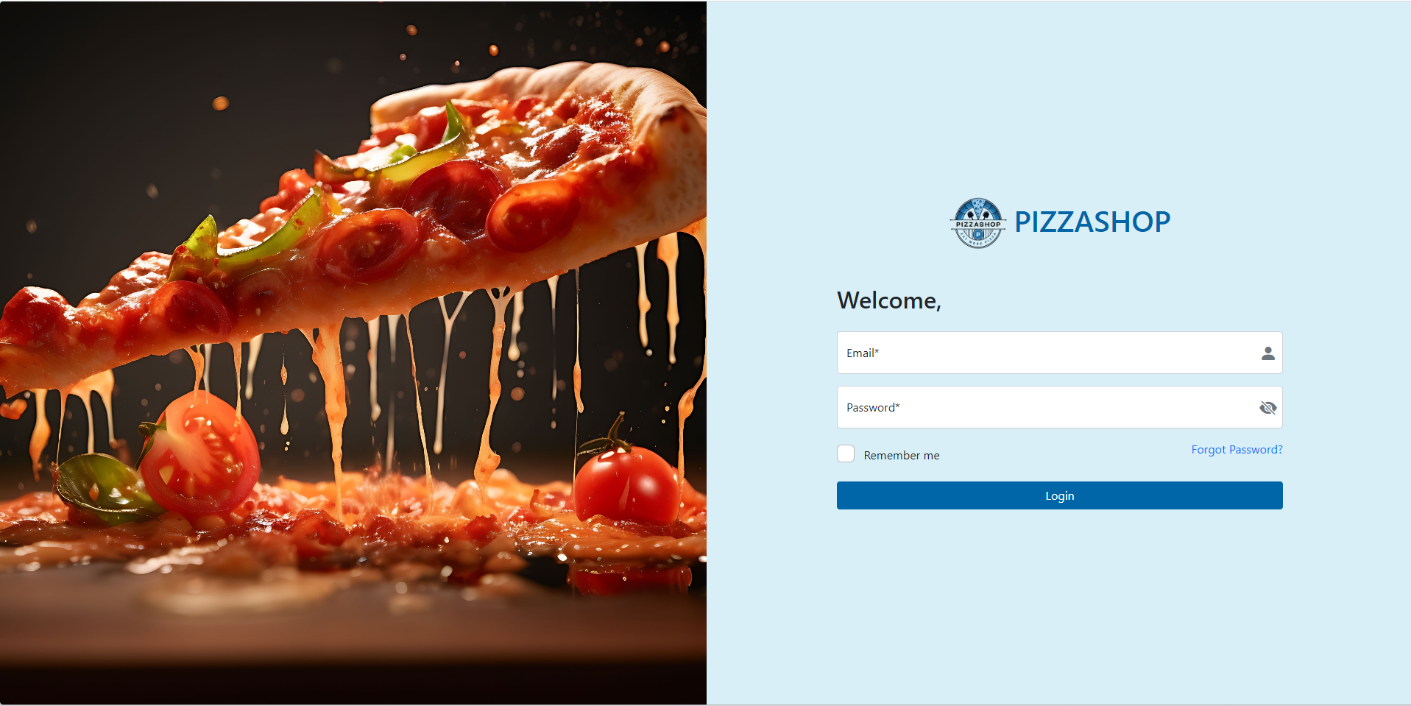


Fig 1 loginScreen

* **Order Management:**

Customers place orders through the system, either through a **dedicated in-store ordering system or via third-party online integrations**.

Orders are automatically categorized based on priority and assigned to the kitchen staff.

Real-time order tracking updates allow kitchen staff and managers to monitor the status.

Once prepared, orders are dispatched to the billing section for final processing.

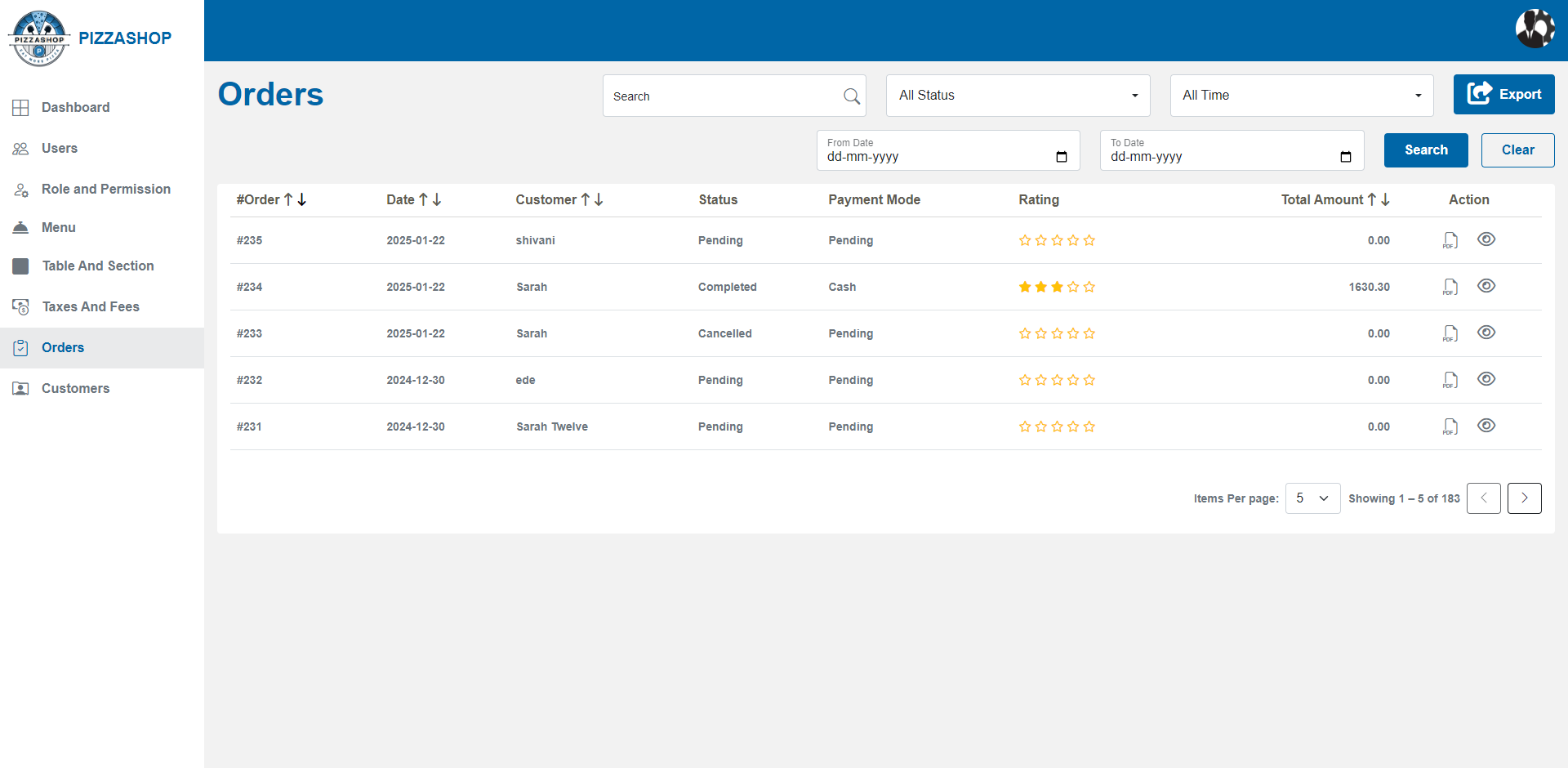


Fig. 2 loginScreen

* **Menu Management:**

Admins have a **centralized dashboard** to update menu items, adjust pricing, and set availability.

The inventory module continuously **monitors stock levels** and **triggers restocking alerts** when predefined thresholds are met.

Suppliers can be integrated into the system, automating procurement based on demand patterns.

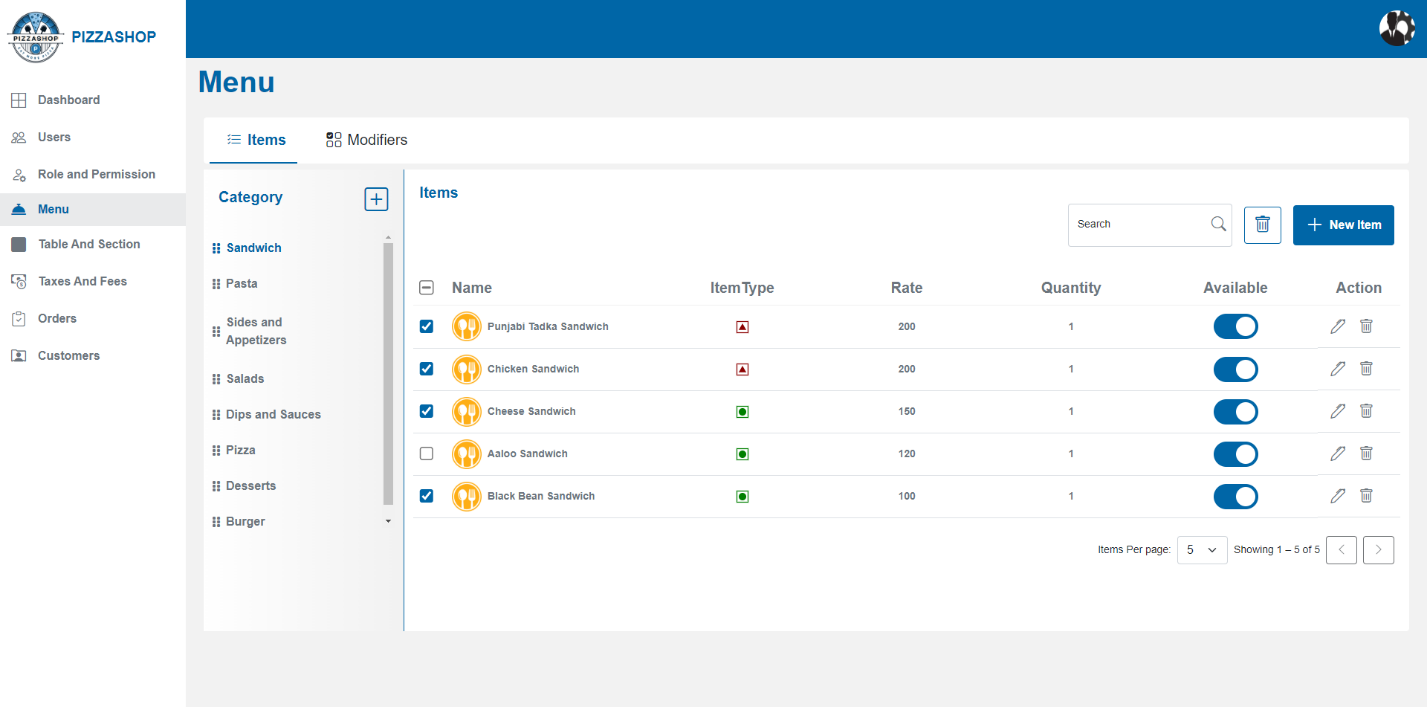


Fig.3 Display the list of all items as per the category

* **Billing & Payment Processing:**

The system generates **automated invoices** linked to order IDs and customer details.

Payments are processed securely through **integrated payment gateways**, ensuring encrypted transactions.

The refund and adjustment module efficiently **manages discrepancies** and issues reimbursements when necessary.



Fig.4 Invoice Pdf

* **Reporting & Analytics:**

The system generates **dynamic reports on sales, inventory trends, and customer behavior**, assisting in decision-making.

Analytics to suggest menu optimization strategies, pricing adjustments, and operational improvements based on data trends

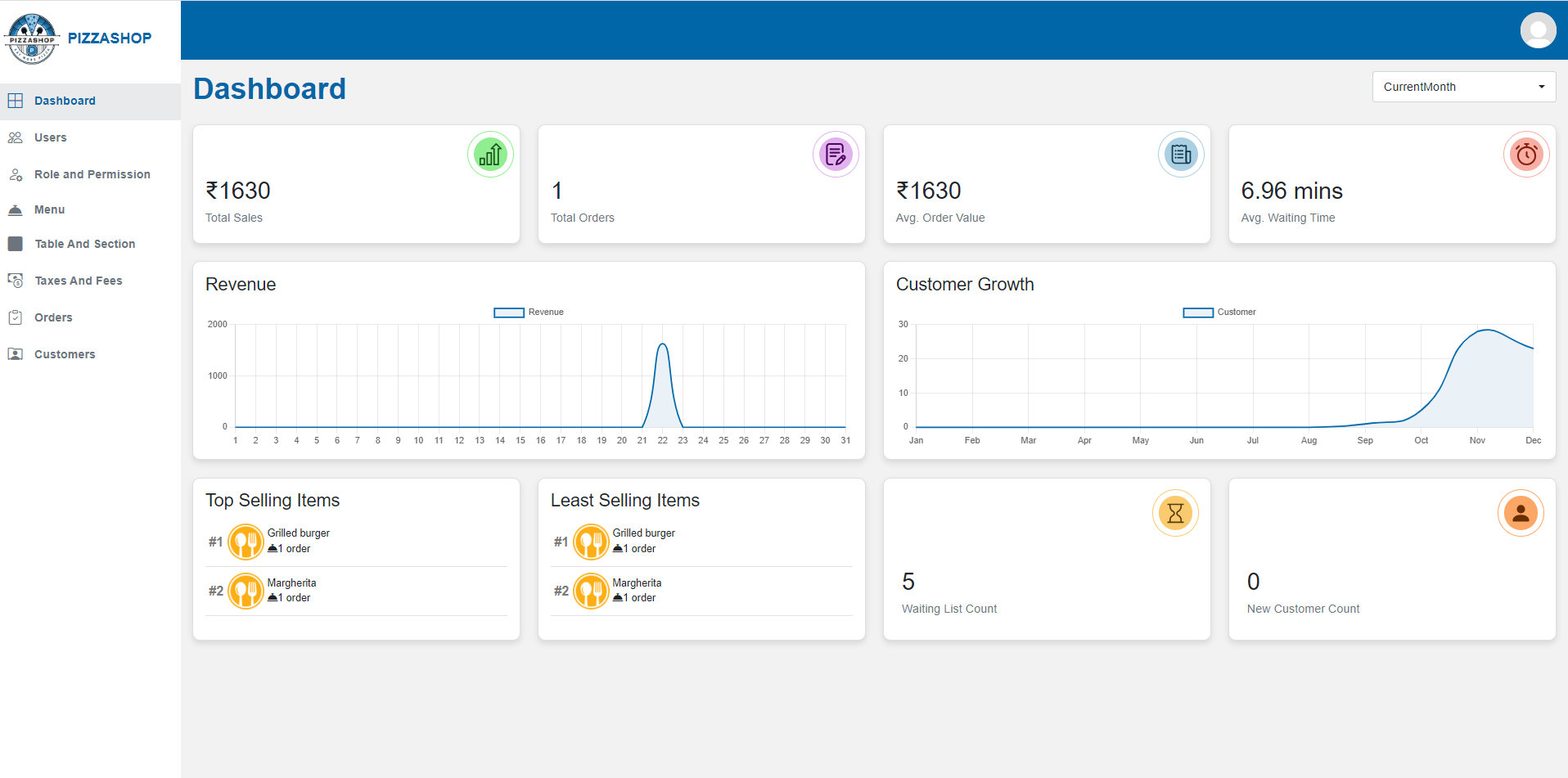


Fig.5 Dashboard page

* **Table Module in Order App :**

The **Table Menu** is central to order management in a restaurant, ensuring that the Account Manager can handle customer requests efficiently, track order progress, and manage table reservations seamlessly.

The Table Menu shows a real-time overview of all available, occupied, and reserved tables in the restaurant.

Each table is displayed with its status (available, selected, running or reserved), the number of seats etc.

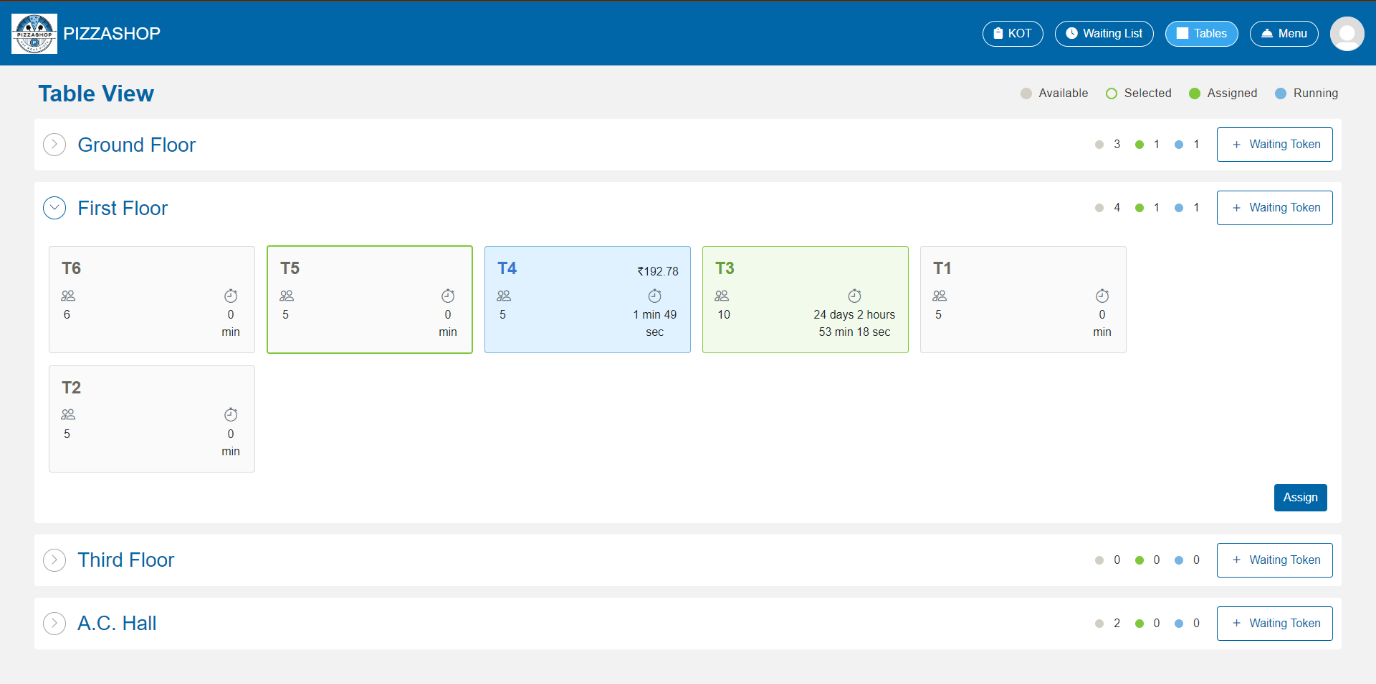


Fig 6 Table Module Order App

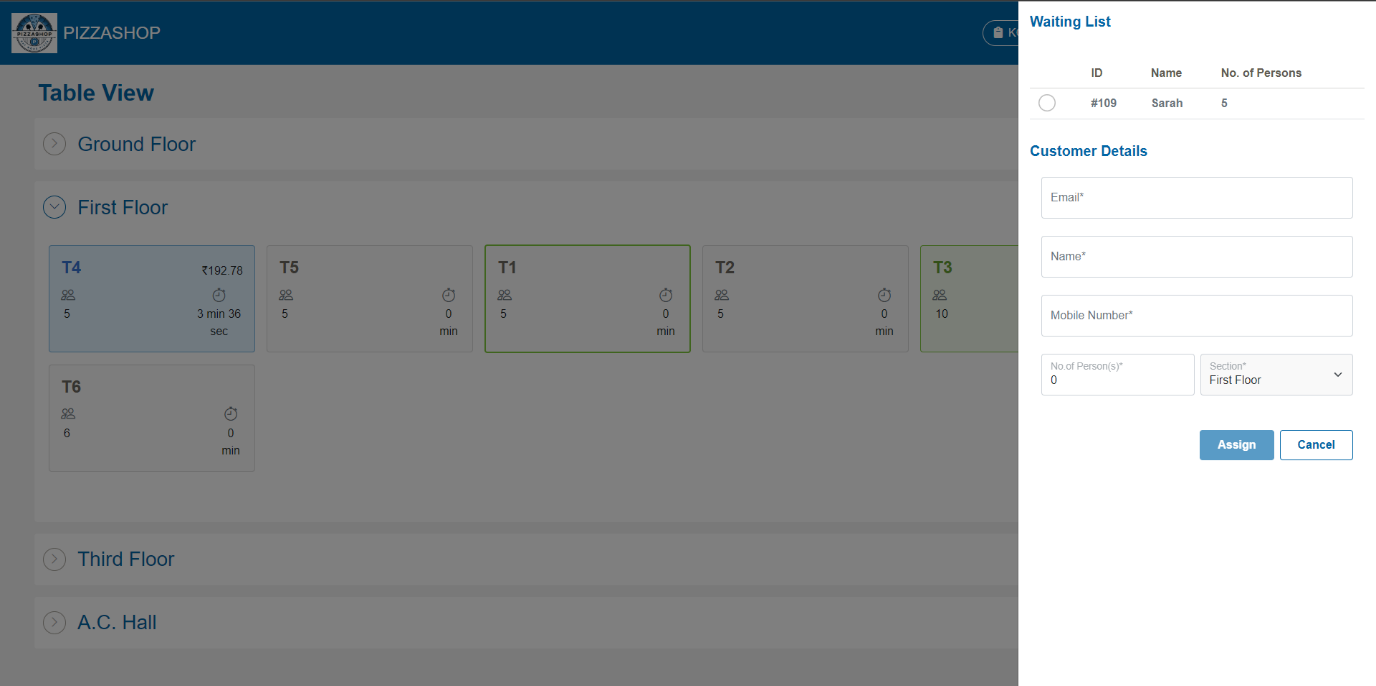


Fig 7 Table Module Order App

* **Menu module in order app:**

Menu contains all types of food categories such as Favourite Items, All, Appetite, Appetizer, Deserts, Pizzas, Pasta, Dips and Sauces, Sand witches etc. On clicking the categories, the respective food items will appear on the screen and According to the categories account manager can add the food items to the table

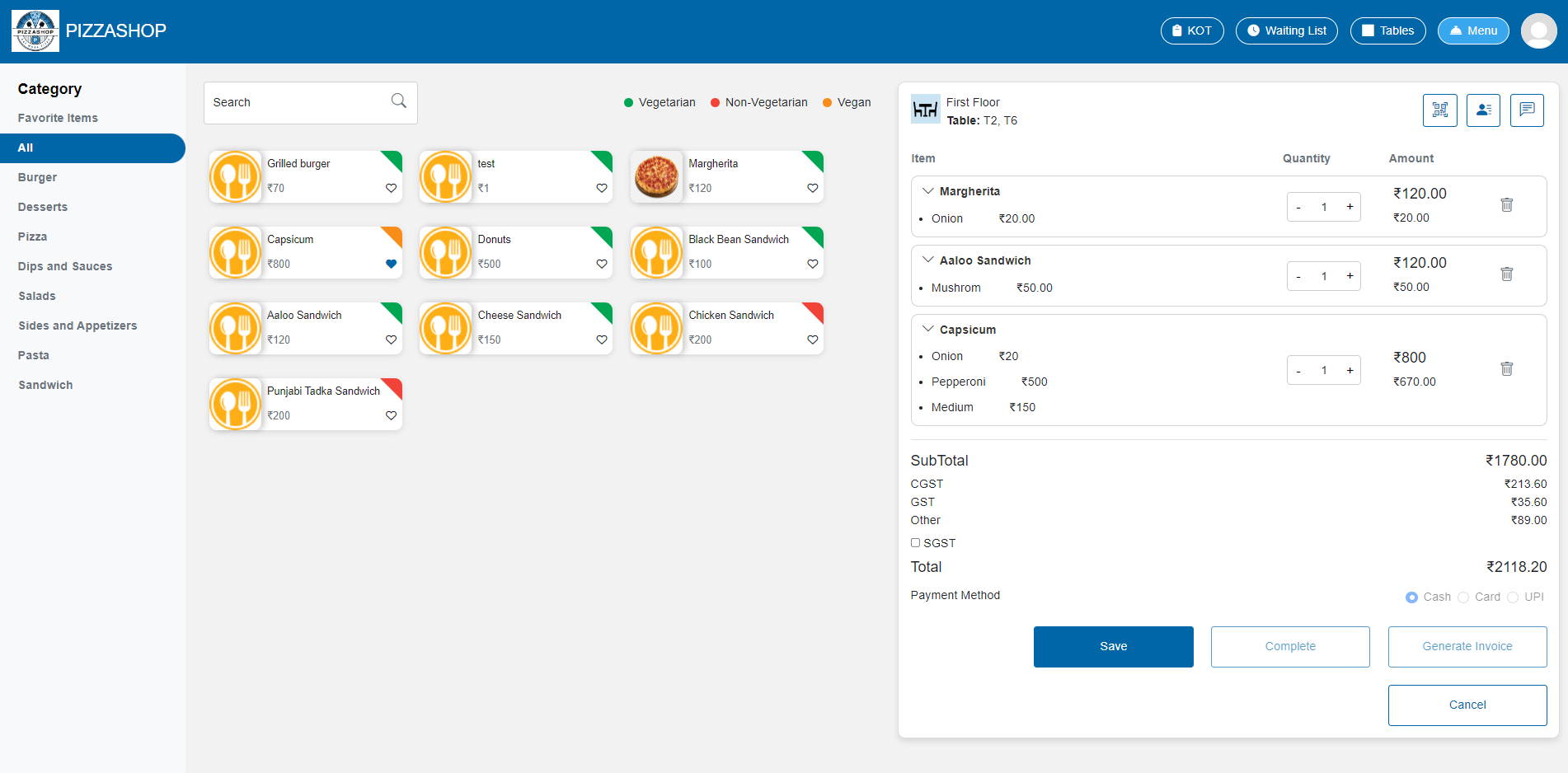


Fig 8 Menu Module Order App

* **Waiting List module in order app**

The **Waiting List** menu in the **Order App** is designed to manage customers who are waiting for a table when the restaurant is full. This menu helps manage customer flow during busy times, improving efficiency and customer satisfaction

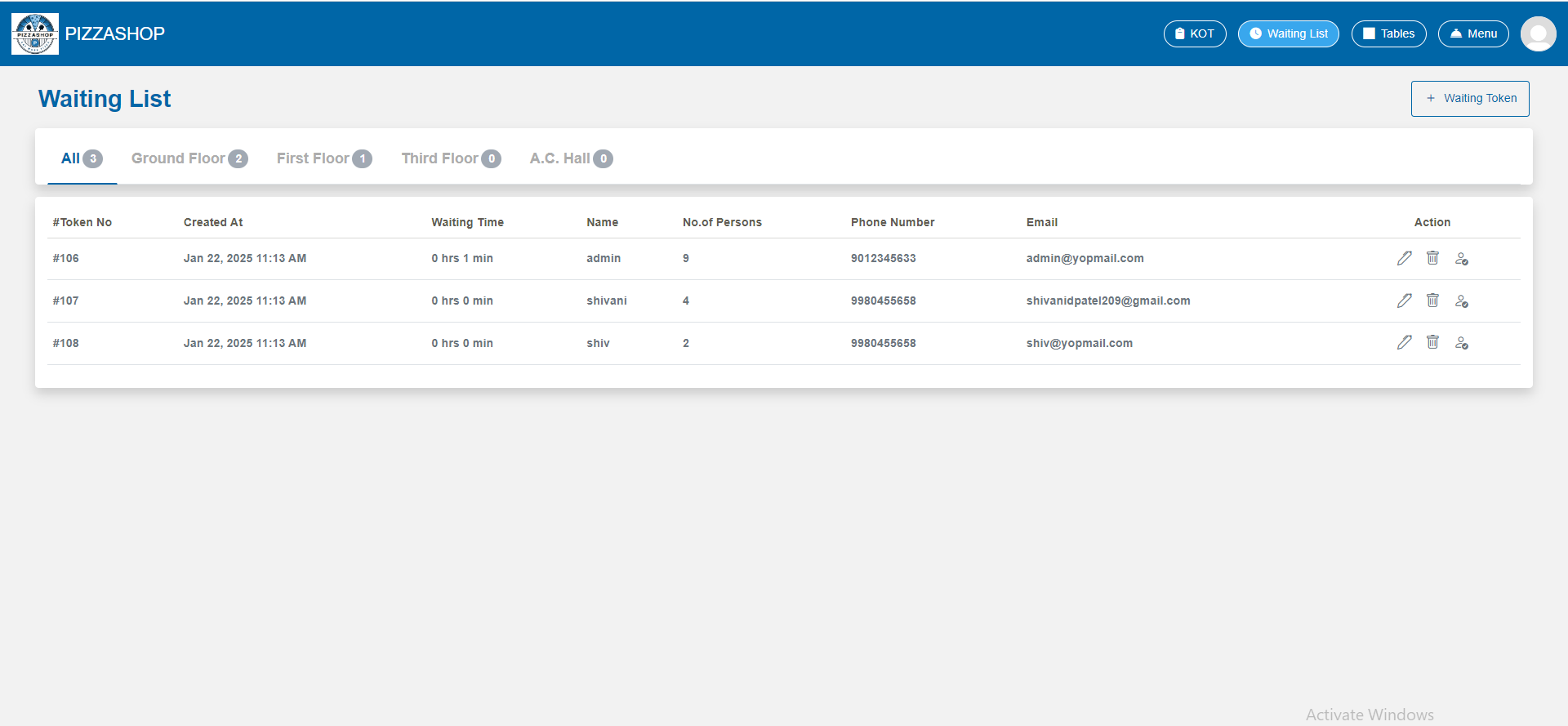


Fig 9 Waiting List Order App

* **KOT**

In KOT, chef can view all the food items ordered by customers, categorized according to the menu sections with status In Progress or Ready as shown in the image below.

The KOTs for each table are displayed in real-time on the Chef Dashboard, showing all incoming food orders. The orders are grouped by categories, such as Pizzas, Burgers, and Desserts, allowing chefs to manage and organize their workload effectively.

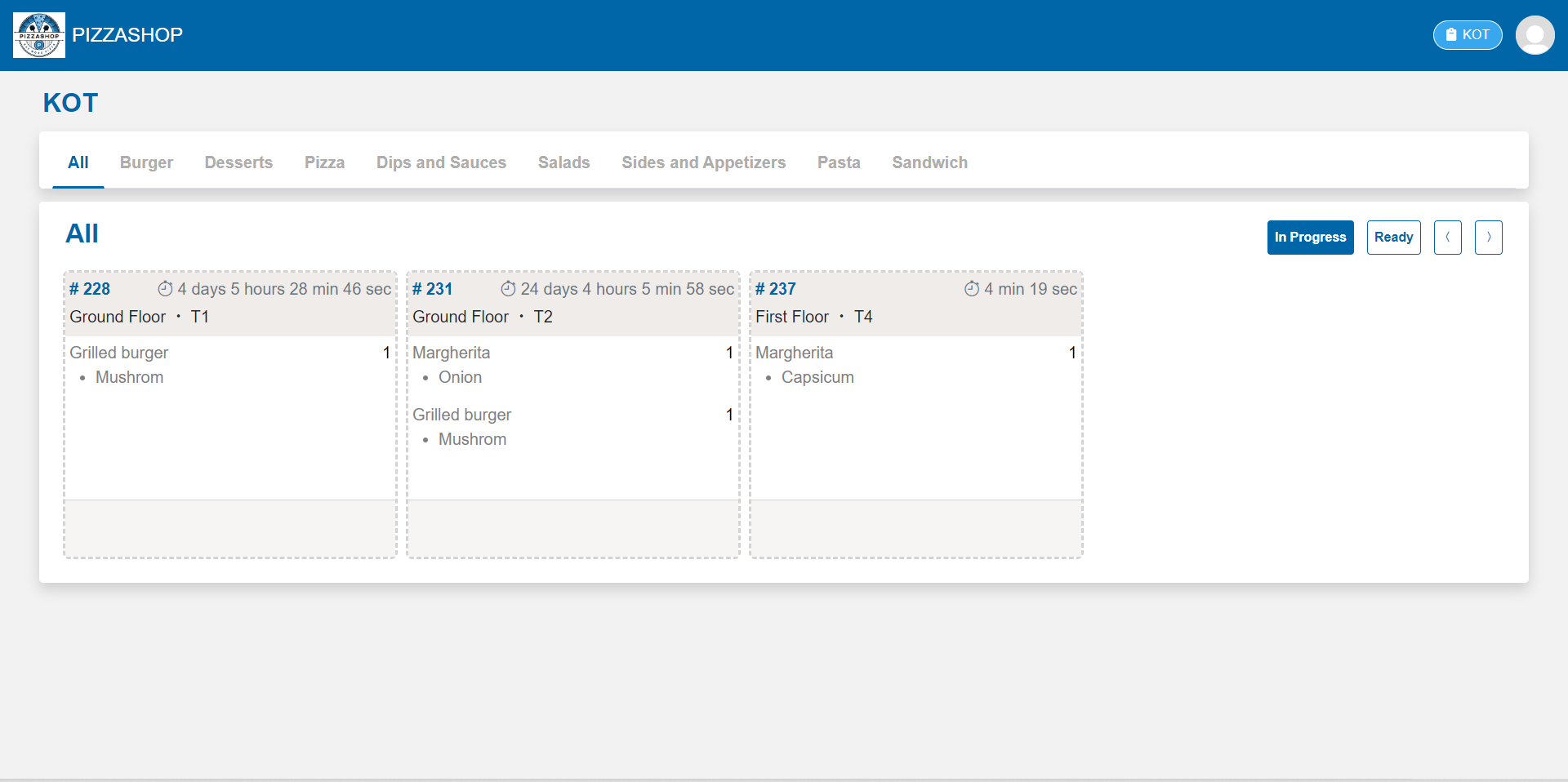


Fig 10 KOT Order App

### 5.2 System Requirements

#### 5.2.1 Hardware Requirements

* **Processor:** Intel Core i5 or higher (or equivalent AMD), ensuring smooth performance for high transaction volumes.
* **RAM:** Minimum **8GB RAM** (16GB recommended for production environments to handle concurrent users efficiently).\
* **Storage:** At least **250GB SSD** (500GB recommended for handling large-scale data operations and backups).
* **Network:** Stable **high-speed internet connection** for cloud-based functionalities and remote access.
* **Peripherals:** Support for **POS hardware** (thermal printers, barcode scanners, touchscreens) for seamless operations.

#### 5.2.2 Software Requirements

* **Operating System:** Compatible with **Windows 10/11, Ubuntu 20.04+, macOS** (for development and deployment environments).
* **Backend Technologies:** **ASP.NET Core 7 MVC** framework for business logic implementation.
* **Frontend Technologies:** Developed using **HTML, CSS, JavaScript, and Bootstrap** for a responsive and user-friendly interface.
* **Database:** **PostgreSQL** for robust, scalable, and structured data storage.
* **Security Protocols:** Implementation of **SSL/TLS encryption** for securing data transmission over the network.

### 5.3 Implementation of Project

#### 5.3.1 Development Environment Setup

* Installed and configured **.NET Core 7 SDK** for backend development.
* Deployed **PostgreSQL** as the primary database, optimizing schema designs for efficiency.
* Set up **Visual Studio** for code development and debugging, along with **Postman** for API testing.
* Configured **Redis caching** to reduce load on the database and improve application performance.
* Integrated **Firebase Cloud Messaging** for real-time updates and push notifications.

#### 5.3.2 Frontend Implementation

* Developed an intuitive and **responsive user interface** using **HTML, CSS, Bootstrap, and JavaScript**.
* Implemented **dynamic user authentication screens**, ensuring seamless login and session management.
* Integrated a **real-time order tracking dashboard**, allowing kitchen staff and managers to monitor order statuses live.
* Developed interactive **menu and inventory management panels** with role-specific functionalities.

#### 5.3.3 Backend Implementation

* Designed a **RESTful API architecture** using **ASP.NET Core 7 MVC**, ensuring modularity and scalability.
* Implemented **business logic for order processing, inventory tracking, and financial transactions**.
* Developed **error-handling and logging mechanisms** to track system failures and performance issues.
* Integrated third-party **payment gateways** for secure and efficient transaction handling.
* Configured **scheduled background tasks** to manage inventory updates and data backups.

#### 5.3.4 Database Implementation

* Designed and structured relational tables for **Users, Orders, Menu, Inventory, Payments, and Analytics**.
* Created **stored procedures and triggers** to automate repetitive tasks like order fulfillment and inventory deductions.
* Optimized database queries to **enhance performance and reduce latency**.
* Implemented **data indexing** to improve query execution speed.

#### 5.3.5 Security Implementation

* Integrated **JWT-based authentication** for secure and tokenized session management.
* Applied **AES encryption** to store and protect sensitive user and payment data.
* Enabled **multi-factor authentication (MFA)** to add an extra layer of security for critical user accounts.
* Implemented **intrusion detection and audit logging** to monitor and track system activities.
* Enforced **strict API rate limiting and validation mechanisms** to prevent cyber attacks.

By following this structured implementation approach, the **Pizza Shop Management System** was developed as a **scalable, secure, and efficient** solution for restaurant operations, ensuring **optimized workflows, enhanced security, and data-driven decision-making**.

## Future Improvements

While the current system effectively handles restaurant-side operations such as order management, table allocation, and kitchen coordination, several key features can be added to improve the system’s functionality and customer engagement. Below are the proposed enhancements:

* **Customer-Facing Interface :**  
  A dedicated interface or mobile app for customers can be developed, allowing them to browse the menu, place orders, and view offers. This will improve customer convenience and reduce dependency on staff for basic interactions.
* **Online Booking & Reservation System :**  
  Integrating a table reservation module will enable customers to book tables in advance. This will reduce wait times, optimize table occupancy, and allow better planning for peak hours.
* **Payment Gateway Integration :**  
  Adding secure online payment options (UPI, cards, wallets) will streamline the billing process, support online orders, and improve customer trust. It also reduces cash handling errors and delays.
* **Multi-Branch Support :**  
  If the restaurant grows into a chain, supporting multiple branches from a centralized dashboard will be critical. This enables standardized reporting and resource allocation.
* **Integration with Food Delivery Platforms :**  
  Linking with platforms like Zomato or Swiggy via APIs will expand reach and boost sales without developing a separate delivery infrastructure.

# Chapter 6

## Conclusion

### 6.1 Summary

### The Pizza Shop Management System has been successfully developed to streamline restaurant operations through structured modules including user authentication, order processing, inventory management, billing, and reporting. By implementing secure JWT-based login mechanisms and role-based access control, the system ensures operational security and integrity. Real-time order tracking, centralized menu control, automated billing, and analytics-driven reporting significantly enhance efficiency, reduce manual workload, and support data-driven decisions.

### However, to align with modern consumer expectations and further enhance the system’s value, several enhancements have been identified for future implementation. These include a customer-facing interface for seamless online ordering, table reservation features to optimize seating management, integrated payment gateways for smooth digital transactions, and loyalty programs to increase customer retention. Additional tools such as feedback systems, marketing automation, and multi-branch support will position the platform as a comprehensive solution capable of supporting both single-location restaurants and scalable franchises.

### With these improvements, the Pizza Shop Management System can evolve from a back-end operational tool into a fully integrated, customer-centric platform that supports both business growth and superior user experience.

# REFERENCE

## Websites

### https://online.visual-paradigm.com

### https://learn.microsoft.com/en-us/aspnet/mvc/

### https://getbootstrap.com/docs/5.0/getting-started/introduction/

### https://stackoverflow.com/

### *https://*[*www.w3schools.com/*](http://www.w3schools.com/)

### *https://*[*www.c-sharpcorner.com/*](http://www.c-sharpcorner.com/)

### https://dotnettutorials.net/