**DevOps Internship Program EBPL-2025**

**Name : HARINI S**

**College Code : 6130**

**College Name : Vivekanandha College of Technology For Women**

**Reg.NO : 613022225011**

**Dept.Name : Agricultural Engineering**

# Project Title: College Event Registration Portal

**Git Hub Project repo url :**

## 1. Project Overview

College Event Registration Portal is a web-based application designed to facilitate the registration and management of events within a college.

## 2. Abstract

The College Event Registration Portal is a web-based application designed to streamline the management of events within a college. It enables participants to view upcoming events, register online, and receive updates. The portal aims to enhance efficiency, accessibility, and communication for event organizers and participants.

## 3. Objective

* In order-taking and billing. - Event: Represents a college event with attributes like date, time, venue, description, and registration details.
* User: Includes participants (students, faculty) and organizers with attributes like name, role, and registration status for events.
* Registration: Represents the act of a user registering for an event, potentially including payment status if applicable.

**4. Problem Statement**

* Manual Registration Processes: Traditional methods like paper forms or email registrations can be time-consuming and prone to errors.
* Limited Accessibility: Participants might find it hard to get event details or register easily.
* Communication Gaps: Organizers might struggle to communicate updates effectively to registered participants.

## 5. Proposed Solution

A responsive, web-based ordering platform that:

* Allows customers to view menus and place orders directly from their devices.
* Sends real-time order notifications to the kitchen.
* Provides an admin dashboard for menu and price updates.
* Offers secure payment integration for hassle-free billing.

## 6. Features

* Event Listing: Display upcoming events with details like date, time, venue, and description.
* Online Registration: Allow participants to register for events online.
* User Management: Manage roles like participants, organizers, and admins.

## 7. Technologies & Tools

* **Frontend:** HTML, CSS, JavaScript (or React.js)
* **Backend:** Node.js, Express.js
* **Database:** MongoDB
* **Optional:** Socket.io for real-time updates, Razorpay/Stripe for payments
* **Tools:** GitHub, VS Code, Postman

## 8. System Architecture

**Flow:**

Customer Device → Frontend UI → Backend API → Database (MongoDB) → Kitchen/Admin Dashboard.

* The backend handles API requests for menu retrieval, order creation, and status updates.
* The frontend communicates with the backend using REST APIs or WebSockets.

## 9. Implementation Steps

1. **Setup Backend:** Node.js + Express + MongoDB.
2. **Create Models:** Menu, Order, User.
3. **Develop APIs:**
   * GET /menu – fetch menu items. o POST /order – create new order.
   * PUT /order/:id – update order status.
4. **Build Frontend:** Responsive HTML/CSS/JS or React interface.
5. **Integrate Payments:** Razorpay or Stripe API.
6. **Testing:** Use Postman & browser testing.
7. **Deploy:** Host backend on Render/Heroku, frontend on Netlify/Vercel.

## 10. Testing

1. **Unit Testing** 
   * Verified backend APIs like /menu, /order, /payment.
   * Example: POST /order → Returns Order Created Successfully.
2. **Integration Testing** 
   * Tested how frontend interacts with backend.
   * Example: Adding items to cart updates MongoDB successfully.
3. **Functional Testing** 
   * User flow tested: Login → Browse Menu → Add to Cart → Checkout → Payment → Kitchen Notification.
4. **User Acceptance Testing (UAT)** 
   * Conducted trials with sample users.
   * Feedback: Easy to use, menu browsing was smooth, real-time order confirmation worked.
5. **Performance Testing** 
   * App tested with 50 concurrent users using JMeter.
   * No downtime, average response time < 2 seconds.
6. **Security Testing** 
   * Checked for SQL injection (not applicable since MongoDB used, but sanitized inputs).
   * Payment gateway uses encryption (Razorpay/Stripe API).

## 11. Results & Discussion

* The system performed efficiently under normal and heavy loads.
* Order management improved with real-time updates, reducing manual errors.
* Customers appreciated the digital interface and faster ordering process.
* Admins found it easy to update menu items without printing costs.
* Minor UI issues (like button alignment on mobile) were fixed after UAT feedback.
* The app proves to be a cost-effective solution for restaurants, especially small to mid-sized ones.
* It enhances the dining experience, reduces waiting times, and simplifies management. With further integration (delivery services, AI recommendations), it can become a full-scale restaurant management platform.

## 12. References

1. W3Schools – *HTML, CSS, JavaScript Tutorials* – [https://www.w3schools.com](https://www.w3schools.com/)
2. MongoDB Documentation – *Database Management* – <https://www.mongodb.com/docs>
3. Express.js – *Web Framework for Node.js* – [https://expressjs.com](https://expressjs.com/)
4. Razorpay API Docs – *Payment Gateway Integration* – <https://razorpay.com/docs>
5. JMeter – *Performance Testing Tool* – [https://jmeter.apache.org](https://jmeter.apache.org/)

## 13. Advantages

* Faster service & reduced wait times.
* Easy menu updates without reprinting costs.
* Reduced order-taking errors.
* Enhanced customer experience with real-time updates.

## 14. Future Enhancements

* Multi-language support for diverse customers.
* AI-based dish recommendations based on customer preferences.
* Integration with food delivery services.
* Voice-enabled ordering for accessibility.

## 15. Conclusion

The Restaurant Menu Ordering App revolutionizes the dining experience by merging technology with restaurant services. By enabling customers to place orders directly and providing the kitchen with real-time updates, the app reduces operational delays, minimizes errors, and improves overall service quality. With future enhancements, it can become a fully integrated restaurant management solution.