**Index**

1. [**Introduction**](#Intro)
2. [**Objectives**](#objectives)
3. [**Problem Statement**](#Synopsis)
4. [**Hardware/ Software Requirements**](#hswreq)

**Introduction**

The thirst for learning, upgrading technical skills and applying the concepts in real life environment at a fast pace is what the industry demands from IT professionals today. However busy work schedules, far-flung locations, unavailability of convenient time-slots pose as major barriers when it comes to applying the concepts into realism. And hence the need to look out for alternative means of implementation in the form of laddered approach.

The above truly pose as constraints especially for our students too! With their busy schedules, it is indeed difficult for our students to keep up with the genuine and constant need for integrated application which can be seen live especially so in the field of IT education where technology can change on the spur of a moment. *Well, technology does come to our rescue at such times!!*

Keeping the above in mind and in tune with our constant endeavour to use Technology in our training model, we at Aptech have thought of revolutionizing the way our students learn and implement the concepts using tools themselves by providing a *live and synchronous eProject learning environment!*

**So what is this eProject?**

eProject is a step by step learning environment that closely simulates the classroom and Lab based learning environment into actual implementation. It is a project implementation at your fingertips!! An electronic, live juncture on the machine that allows you to

* Practice step by step i.e. laddered approach.
* Build a larger more robust application.
* Usage of certain utilities in applications designed by user.
* Single program to unified code leading to a complete application.
* Learn implementation of concepts in a phased manner.
* Enhance skills and add value.
* Work on real life projects.
* Give a real life scenario and help to create applications more complicated and useful.
* Mentoring through email support.

The students at the centre are expected to complete this eProject and send complete documentation with source code to eProjects Team

Looking forward to a positive response from your end!!

**Objectives of the project**

The Objective of this program is to give a sample project to work on real life projects. These applications help you build a larger more robust application.

The objective is not to teach you the concepts but to provide you with a real life scenario and help you create applications using the tools.

You can revise them before you start with the project.

It is very essential that a student has a clear understanding of the subject.

Kindly get back to eProjects Team in case of any doubts regarding the application or its objectives.

**Background**

**"LuxuryStay Hospitality"**, a high-end hotel chain known for its exceptional service and luxurious accommodations. With a network of hotels in various locations, LuxuryStay aims to revolutionize its operational efficiency by implementing a cutting-edge Hotel Management System (HMS).

LuxuryStay envisions a modern, scalable, and user-friendly Hotel Management System to transform its operations. The system will streamline workflows, enhance guest experiences, and provide insightful data for strategic decision-making.

**Functional Requirements**

**User Management:**

**Admin Dashboard:** An admin panel to manage user roles and permissions.

**Staff Profiles:** Ability to create, modify, and deactivate staff accounts with different access levels (manager, receptionist, housekeeping, etc.).

**Guest Profiles:** Creation and management of guest profiles with details like personal information, contact, preferences, etc.

**Room Management:**

**Room Inventory:** Maintain a comprehensive inventory of rooms with details on room types, availability, status (cleaning, occupied, available), and pricing.

**Room Booking:** Allow staff to reserve rooms, check availability, assign rooms to guests, and manage bookings efficiently.

**Room Status Updates:** Enable real-time updates on room status (cleaning, maintenance, vacant, occupied) for better coordination among staff.

**Reservation and Check-in/out:**

**Reservation System:** Facilitate booking management, allowing guests to make reservations online or through staff, providing confirmation details.

**Check-in/Check-out**: Smooth check-in and check-out processes with automated procedures for room allocation, key issuance, billing, and updating room status.

**Billing and Invoicing:**

**Billing System:** Generate accurate bills based on room rates, additional services (food, laundry, etc.), and duration of stay.

**Invoicing:** Provide guests with detailed invoices that can be printed or emailed, including breakdowns of charges.

**Housekeeping and Maintenance:**

**Housekeeping Management:** Allow housekeeping staff to view room status, schedule cleaning tasks, mark tasks as completed, and report maintenance issues.

**Maintenance Requests:** Enable guests or staff to report maintenance issues and track their resolution status.

**Reporting and Analytics:**

**Reporting Dashboard:** Provide management with insights via customizable reports on occupancy rates, revenue, guest feedback, etc.

**Analytics:** Utilize data analytics to forecast demand, optimize pricing, and improve service offerings.

**Security and Compliance:**

**User Authentication:** Implement secure login/logout mechanisms for staff and guests with role-based access control.

**Feedback and Guest Services:**

**Feedback Mechanism:** Provide a way for guests to leave feedback and ratings, helping in improving services.

**Additional Services:** Allow guests to request additional services like room service, wake-up calls, transportation, etc.

**System Administration:**

**System Settings:** Admin controls for system configurations, including setting room rates, defining policies, managing taxes, etc.

**System Notifications:** Alerts and notifications for staff regarding bookings, maintenance requests, and other critical updates.

**Non-Functional Requirements**

**Performance:**

**Response Time:** The application should respond to user interactions within 1-2 seconds for most operations.

**Scalability:** The system should be able to handle a growing number of users and data without significant performance degradation.

**Concurrent Users:** The application should support hundreds of concurrent users without performance bottlenecks.

**Security:**

**Data Encryption:** All sensitive user data, including passwords and personal information, must be securely encrypted during storage and transmission.

**Authentication:** User authentication should be secure and use industry-standard practices to prevent unauthorized access.

**Authorization:** Access control must be implemented to ensure users can only access their own data or public data, as per their settings.

**Privacy:**

**Data Privacy:** The application must comply with data privacy regulations such as GDPR, ensuring user data is handled and stored with care.

**User Consent:** Users should have control over the data they share and provide informed consent for data processing and sharing.

**Reliability:**

**Uptime:** The application should aim for a minimum of 99% uptime, with scheduled maintenance communicated in advance.

**Data Backup:** Regular automated data backups must be performed to prevent data loss in case of system failures.

**Usability:**

**User Interface Design:** The application should have an intuitive, user-friendly interface with consistent navigation and a responsive design that works on various devices.

**Accessibility:** The application must adhere to accessibility standards (e.g., WCAG) to ensure it is usable by individuals with disabilities.

**Compatibility:**

**Cross-Browser Compatibility:** The application should function correctly on popular web browsers, including Chrome, Firefox, Safari, and Edge.

**Mobile Compatibility:** The application should be responsive and work well on various mobile devices and screen sizes.

**Scalability:**

**Horizontal Scalability:** The architecture should support horizontal scaling to accommodate increased user loads as the user base grows.

**Performance Monitoring:**

**Logging and Monitoring:** The system should include logging and monitoring tools to track application performance, errors, and user activity for debugging and analysis.

**Testing and Quality Assurance:**

**Test Coverage:** A comprehensive test suite should be maintained, covering unit testing, integration testing, and end-to-end testing.

**Security Testing:** Regular security assessments, including penetration testing, must be conducted to identify vulnerabilities.

**Documentation:**

**User Documentation:** Provide user guides, FAQs, and tutorials to help users understand and navigate the application.

**Developer Documentation:** Maintain developer documentation to assist in further development and maintenance.

**Video:** Provide video displaying complete working of the application.

**[Hardware/ Software Requirements](#hswreq)**

**Hardware**

* A minimum computer system that will help you access all the tools in the courses is a Pentium 166 or better
* 128 Megabytes of RAM or better
* Windows 2000 Server (or higher if possible)

**Software**

Use software as per your requirement

* Windows OS /MongoDB/Express/React/Node.js/Notepad