

# **SIX MONTHS INDUSTRIAL TRAINING ON 'MERN STACK' BOOK APPLICATION STORE**

**EXCELLENCE  
TECHNOLOGY**

**Chandigarh**

**Submitted by**

**Amit Kumar**

**Under Supervision of**

**Internal Supervisor(s)**

**External Supervisor  
Trainer**

<b>INDEX</b>			
<b>S.no.</b>	<b>Title</b>	<b>Page no.</b>	<b>Signature</b>
1	Certificate of Company	3	
2	About the Company	3-10	
3	About the Project	12-16	
4	Objective of the Project	16-18	
5	Hardware and Software Requirements	19-22	
6	Future Scope of Work	22-51	
7	Conclusion	51-53	

## About the Company



### ABOUT US

**EXCELLENCE TECHNOLOGY (ET)** is India based leading strategic IT Company offering integrated IT solutions with the vision to provide Excellence in software solution. We at EXCELLENCE TECHNOLOGY bring innovative ideas and cutting edge technologies into business of customers. EXCELLENCE TECHNOLOGY is having rich experience in providing high technology end to end solutions in **MOBILE APP AND WEB DEVELOPMENT**.



### PHILOSOPHY

- ✓ To impart hard-core practical quality training among students/developers about latest technologies trending today.
- ✓ To share knowledge of information security and create awareness in the market. The solution to clients' as per the International standard practices and governance.
- ✓ To support good business practices through continual employee training and education
- ✓ To equip a local team with a strong knowledge of international best practices and international expert support so as to provide practical advisories in the best interests of our clients

## OUR MISSION



To Protect Client's information system by Detecting Threats, Deterring Attacks, and provide best solution in the ever changing Info - Security Challenges. We accomplish this by integrating our People, Process, and Technology in the most efficient way to increase value for all stakeholders

## OUR SERVICES

- ✓ **RISK Management Services**
- ✓ **Quality Control**
- ✓ **Business Process Re-Engineering**
- ✓ **Network Risk Analysis**
- ✓ **Software Testing**
- ✓ **Mobile Application Testing**
- ✓ **Wireless Penetration Testing**
- ✓ **Network Penetration Testing**
- ✓ **Application Security Testing**



### **OUR SERVICES IN SOFTWARE DEVELOPMENT**

*We are proficient in all platforms of software Development practices — Agile, SCRUM, Lean, Waterfall, Prototype, Incremental, Iterative, and V-Mo*

AMIT KUMAR

With the EXCELLENCE TECHNOLOGY experience the incredible services such as agile software development and the problems related to outsourcing. We comprise of the team of experienced and professionals members who with their skills efficiently get the job done and innovatively help you to transform your ideas into the successful business.

## Why Choose Us?

- ✔ **Hundreds of Clients & Nearly a Decade of Experience**
  - ✔ **Goal Oriented, ROI-Driven Focus**
  - ✔ **A Streamlined / Quality-Driven Process**
  - ✔ **Talented Designers & Expert Developers**
  - ✔ **Our Websites & E-marketing Platforms are Easy to Manage**
  - ✔ **We Are Dedicated to Our Clients' Success**
- 
- ✔ **EXCELLENCE TECHNOLOGY** is steadfast to undertake the projects cutting edge to technology competence and know-how abilities. The project execution is held with dedication and responsibility to perform our best with the essence of knowledge, creativity and skills to the utmost and efficiently.
  - ✔ At **EXCELLENCE TECHNOLOGY**, we have competence to expand and adjust as per client specific requirements.
  - ✔ **Skilled Workforce:** At EXCELLENCE TECHNOLOGY you deal with the highly professional and proficient employees.
  - ✔ **Cost Efficiency:** We help you to reduce the unnecessary investment and ask for the reasonable amount of money.



- ✔ **Quality Of the Product:** Our software service sector has been maintaining the highest international standards of quality.
- ✔ **Infrastructure:** Well organized team and tools to handle the projects with responsible approach Hardware, Software, Networking, Voice, Conferencing, disaster recovery all infra all you need for international projects.
- ✔ **Ongoing Involvement:** EXCELLENCE TECHNOLOGY products are “built for change” as we are well responsive that the necessity to improve a Web solution generally arises even before the solution is out of the door. We delivers long-term product enhancement if desired.
- ✔ **Partnership:** EXCELLENCE TECHNOLOGY considers every client a partner. From the initial stages, you are closely involved into the procedure of technical classification, development, and testing.

## OUR CLIENTS



*Zimmerman Metals Inc.*



Irrigation & Landscaping









## KEY PROFESSIONALS:

In addition to a panel of eminent consultants and advisors, we have a dedicated pool of trained Developers and Trainer, investigators, working under the guidance of professional

managers. **“A Ship is as good as the crew who sail her.”** Our Technical team of professionals handling, designing & delivering of projects has a strong presence in the North India & the US. Our engineers are already working on the latest technologies like **I-Phone & Android** Applications, **Robotics, VLSI-VHDL**, Embedded System, Networking and **Cloud computing**. Some of the key professionals and advisors are listed below:

1. Mr. Deepak Kashyap: (Branch Manager)

He is the backbone of Excellence Technology and a man with more than 10 year rich practical experience who believes in taking up new ventures and projects. Head-IT, Global Award Winner from India for AGILE and OPEN GRAPH implementation in **Sudan Taxation Chamber**. Divisional Award and Star Award Winner in Year 2011 for his exemplary work in process improvement for IT Service Delivery Domains. MASTERS in Computer applications and Certified from PTU Certification. Holds total of 10 Years of rich experience including 5 Years in Information Security Implementation, Maintenance and Auditing and initial over 10 years' experience in Project Management, Client Relationship Management and Server, Desktop and IT Service Delivery.

2.Ms. Sunita Thakur : (Dean Academics)

A Woman who believes that “Honour Time & Place, then you will be honoured.” She has more than 4 years solid industrial experience in a software companies & is very dashing and innovative in her technical approach. MCA, Diploma in Information Technology, expertise in search engine optimization and web designing.

3.Mr. Gurpreet Singh : (Technical Head – MERN STACK & GIT HUB)

A man who strongly feel that “Nothing is Impossible”. A very committed team leader who has been professionally attached with Multinational companies for more than 18 years and has lead the marketing teams in all states of North India. MCA (Engineering), in computer Science PTU Approved trained and Certified. Successfully MCA master degree Training program for a leading Technology. Holds total of 5 Years of rich experience including 3 Years in Software developer, Maintenance and Auditing and initial over 4 years experience in Project Management, Client Relationship Management and IT Service Delivery.

#### **4. Miss. Anjali Sharma : (Sr. Counselor)**

A woman who believes that “Challenges are what make life interesting and overcoming them is what makes life meaningful.” She has more than 3years experience in business development. B.A. – (Punjab University) Lead Consultant - Due Diligence, Strategy, Operations, Business Continuity, Risk Management, Mergers & Acquisitions.

#### **5. Mr. Akshit Dogra : (Mechanical Head)**

A man believes that “don’t wait for extra ordinary opportunities, seize common occasions and make them great.” He has more than 4 years experience in marketing field. B. Tech (Mech), Engineer and Value, is a approved by AICTE (HPTU). He is having total 5 years of experience in 3 years of experience in Autocad, Solidworks , Catia, Pro-e, Ansys etc.

#### **6. Mr. Paramveer Singh : (Mobile Application & Web Developer)**

B. Tech (CSE) – IKG-PTU, Expertise in Reviewing current systems, Presenting ideas for web development soft wares, including Application software, Working closely with analysts, Senior Developers, Programmers, designers and staff, Producing detailed specifications and writing the program codes, Testing the product in controlled, real situations before going live, Preparation of training manuals for users and Maintaining the systems once they are up and running

**Corporate Office: SCO 80-82, Third Floor**  
**Sector-34, Chandigarh, 160022 .**  
**PH.:0172-4698755,93177-88822 ,93562-55522**  
**E-MAIL: excellencehead@gmail.com,**  
**URL: [www.excellencetechnology.in](http://www.excellencetechnology.in)**

### **About Project**

#### **1. Introduction**

Project Name: **Hotel-Hub Room Booking Website**

Developer: **AMIT KUMAR**

The **Hotel-Hub Room Booking Website** is a full-stack web application developed using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**. This platform allows users to search, book, and manage hotel rooms online with ease. The system provides functionalities for both customers and hotel administrators, ensuring an efficient and seamless booking experience.

## 2. Objectives

- Develop a responsive and user-friendly hotel booking website.
- Implement secure authentication and authorization.
- Provide real-time room availability and booking management.
- Enable online payment integration for room reservations.
- Implement an admin panel for hotel management.

## 3. Technology Stack

- **Frontend:** React.js, Vite, Bootstrap & CSS
- **Backend:** Node.js, Express.js
- **Database:** MongoDB with Mongoose ORM
- **Authentication:** JSON Web Token (JWT), bcrypt.js
- **Payment Integration:** Stripe API
- **Deployment:** Vercel (Frontend), Render/Heroku (Backend), MongoDB Atlas (Database)

## 4. Features

### 4.1 User Features

- **User registration and login**
- **Search and filter hotels by location, price, and name**

- **View hotel details and available rooms**
- **Book rooms with online payment options**
- **View booking history and manage reservations**

#### 4.2 Owner Features

- **Owner registration and login**
- **Create Hotel details**
- **Update and Delete Room**
- **Create and Manage Room Availability**
- **View Booking Details**

1.

#### 4.3 Admin Features

- **Secure admin panel access**
- **Add, update, and remove hotel listings**
- **Manage room availability and bookings**
- **View customer details and transaction history**

### 5. System Architecture

The system follows an MVC architecture:

- **Frontend (React.js):** Handles UI, API calls, and state management.
- **Backend (Node.js, Express.js):** Handles business logic, authentication, and API endpoints.
- **Database (MongoDB):** Stores user, owner & admin data, hotel listings, and booking details.

### 6. Database Design

**Collections:**

- **Users** (User ID, Name, Email, Password, Role)
- **Owners** (Owner ID, Name, Email, Password, Role)
- **Admins** (Admin ID, Name, Email, Password, Role)
- **Hotels** (Hotel ID, Name, Location, Description, Images, Price, Amenities)

- **Rooms** (Room ID, Hotel ID, Type, Price, Availability, Features)
- **Bookings** (Booking ID, User ID, Room ID, Check-in, Check-out, Payment Status)

## 7. Implementation Details

### 7.1 Frontend

- Developed using **React.js** with **Vite** for fast builds and better performance.
- State management handled using **Redux Toolkit**.
- UI components designed with **Bootstrap & CSS**.
- API requests handled using **Axios**.

### 7.2 Backend

- Built using **Node.js** and **Express.js**.
- Routes secured using **JWT authentication**.
- Data stored in **MongoDB Atlas** with **Mongoose ODM**.

### 7.3 Payment Integration

- Integrated **Stripe API** for secure transactions.
- Users receive email confirmations upon successful bookings.

## 8. Challenges and Solutions

- **Data Security:** Implemented **bcrypt.js** for password hashing and **JWT** for secure authentication.
- **Scalability:** Used **MongoDB Atlas** for cloud-based data storage and optimized API performance.
- **UI Responsiveness:** Designed mobile-first UI with **Tailwind CSS** for a seamless experience across devices.

## 9. Future Enhancements

- Implement real-time chat support for customer inquiries.
- Add a review and rating system for hotels.
- Enhance search functionality with AI-based recommendations.

- Develop a mobile application version.

## 10. Conclusion

The **Hotel Room Booking Website** is a fully functional web application that simplifies the booking process for users while providing hotel administrators with an efficient management system. With further enhancements, it can become a robust and scalable platform for online hotel reservations.

AMIT KUMAR

## **The Importance of Hotel-Hub Room Booking Website: -**

The Hotel-Hub Room Booking System plays a crucial role in transforming the hospitality industry by streamlining the hotel booking process for both customers and hotel administrators. Below are the key reasons why this project is important:

### **1. Expanding Accessibility to Hotel Bookings**

In today's digital world, traditional hotel booking methods are being replaced by online systems. The Hotel-Hub platform allows users to book rooms from anywhere at any time, making the process more convenient.

- **Global Reach:** Users can search and book hotel rooms from anywhere in the world.
- **Convenience:** No need for in-person reservations; users can book with just a few clicks.

### **2. Cost-Effectiveness for Hotel Owners**

For hotel owners, running an online booking platform is more cost-effective than relying solely on offline reservations.

- **Lower Overheads:** Reduces the need for manual staff to handle reservations.
- **Better Inventory Management:** Real-time updates on room availability and bookings.

2.

### **3. Enhanced User Experience**

A seamless and user-friendly interface significantly improves the hotel booking experience:

- **Search and Filter Options:** Users can filter rooms based on price, amenities, and availability.
- **Personalized Recommendations:** Based on user preferences and booking history.
- **Customer Reviews and Ratings:** Helps users make informed decisions.

### **4. Efficient Room and Booking Management**



For hotel administrators, managing room bookings becomes easier and more efficient:

- **Automatic Room Availability Updates:** As rooms are booked or canceled, availability is updated instantly.
- **Order Tracking:** Users can track the status of their bookings in real-time.
- **Revenue Analytics:** Hotels can analyze booking trends to optimize pricing and services.

## 5. Integration with Payment Systems

The platform supports multiple payment gateways, ensuring smooth and secure transactions:

- **Secure Payments:** Integration with trusted payment processors minimizes fraud risks.
- **Multiple Payment Options:** Users can pay using credit/debit cards, UPI, PayPal, etc.

## 6. Providing a Platform for Small Hotels and Guesthouses

Many small hotels and guesthouses struggle with digital presence. Hotel-Hub provides an opportunity for them to reach a wider audience.

- **Direct Listings:** Hotel owners can list their properties and manage their rooms.
- **Better Exposure:** Helps independent hotels compete with larger chains.

## 7. Promotion and Marketing Opportunities

Hotel-Hub enables hotel owners to market their rooms effectively:

- **Discounts and Offers:** Hotels can provide exclusive deals to attract customers.
- **Targeted Advertising:** Users receive personalized promotions based on their preferences.
- **Social Media Integration:** Users can share their experiences, boosting hotel visibility.

## 8. Sustainability

Moving towards an online booking system reduces paperwork and promotes sustainable business practices:

- **Digital Booking Records:** Eliminates the need for printed receipts and records.
- **Reduced Energy Consumption:** Less reliance on call centers for manual bookings.

## 9. Data-Driven Insights for Continuous Improvement

The system collects valuable data to help hotel owners improve their services:

- **Customer Behavior Analysis:** Identifies booking trends and peak seasons.
- **Feedback Mechanism:** Helps hotels improve based on customer reviews.

## 10. Scalability

As the demand for online hotel bookings grows, the Hotel-Hub system can easily scale:

- **Global Expansion:** The platform can onboard more hotels from different locations.
- **Increased Capacity:** Easily accommodates a growing number of users and bookings.

## Conclusion

The Hotel-Hub Room Booking System revolutionizes the hotel industry by offering a seamless, secure, and efficient online booking experience. It benefits both customers and hotel owners by enhancing accessibility, reducing costs, and improving management. As online booking trends continue to grow, platforms like Hotel-Hub will become indispensable in the hospitality sector.

- 3.
- 4.
- 5.
- 6.
- 7.
- 8.

## **System Requirements**

### **Problem statement**

In today's digital era, the increasing prominence of the internet has transformed the economic landscape, placing emphasis not only on traditional product offerings but also on the provision of services. The level of service provided has become a critical determinant of success in the market. To enhance accessibility and convenience for users, we aim to design a chat application that enables seamless communication and interaction without the need for physical presence. The objectives of this chat application are outlined as follows:

To furnish comprehensive information about the platform, facilitating improved user understanding and engagement.

To cater to individuals with time constraints by offering a convenient online communication platform.

To provide diverse payment options, including cash, cheque, and DD, to accommodate varying user preferences.

To offer detailed product descriptions and related information, such as pricing and specifications, ensuring informed decision-making.

To optimize time and cost efficiency for users by streamlining communication processes.

To enable remote communication and interaction through multiple payment methods, including cash, DD, and debit cards.

### **Processing environment**

#### **Hardware requirements:**

##### **Windows**

- Microsoft® Windows® 7/8/10 (64-bit)

- 2 GB RAM minimum.
- 2 GB of available disk space minimum,
- 1280 x 800 minimum screen resolution

## **Linux**

- 64-bit distribution capable of running 32-bit applications
- 2 GB RAM minimum, 4 GB RAM recommended
- 2 GB of available disk space minimum
- 1280 x 800 minimum screen resolution

## **Software requirements:**

To develop a MERN (MongoDB, Express.js, React.js, Node.js) stack project, you'll need a set of software tools and technologies. Here's a breakdown of the requirements:

**Operating System:** You can develop MERN stack applications on Windows, macOS, or Linux. Choose the one you're most comfortable with.

**Text Editor or IDE:** You'll need a text editor or integrated development environment (IDE) to write your code. Popular choices include Visual Studio Code, Sublime Text, Atom, or WebStorm.

**Node.js and npm (Node Package Manager):**

- **Node.js:** This is a JavaScript runtime environment that executes JavaScript code outside of a web browser. You can download and install it from the official Node.js website.
- **npm:** npm comes bundled with Node.js and is used to install packages and manage dependencies for your project.

**MongoDB:** MongoDB is a NoSQL database used in the MERN stack for storing data. You can download and install MongoDB Community Edition from the official MongoDB website.

**Express.js:** Express.js is a web application framework for Node.js. It provides a robust set of features for building web and mobile applications. You'll use npm to install Express.js as a dependency in your project.

**React.js:** React.js is a JavaScript library for building user interfaces. You can set up React.js using Create React App, a tool built by Facebook for quickly setting up React projects. Install it globally using npm.

**Additional Libraries and Tools:**

**React Router:** For client-side routing in React applications.

**Axios or Fetch API:** For making HTTP requests from your React application to your Express.js backend.

**Mongoose:** If you're using MongoDB with Node.js, Mongoose is a popular library for MongoDB object modeling and provides a straightforward schema-based solution for modeling your application data.

**nodemon:** It's a development tool that automatically restarts the node application when file changes are detected. It's handy for the development process.

**Concurrently:** To run both the server-side and client-side development servers concurrently.

**Version Control:** Using a version control system like Git is highly recommended for managing your project's source code. You can use GitHub, GitLab, or Bitbucket for hosting your Git repositories.

Remember to regularly update your software packages and dependencies to benefit from the latest features, bug fixes, and security patches. Additionally, refer to the documentation and resources provided by each technology for more in-depth guidance on installation and usage.

## Solution strategy

### Object Oriented Design

Object oriented design works around the entities and their characteristics instead of functions involved in the software system. This design strategies focuses on entities and its characteristics. The whole concept of software solution revolves around the engaged entities.

Let us see the important concepts of Object Oriented Design:

- **Objects** - All entities involved in the solution design are known as objects. For example, person, banks, company and customers are treated as objects. Every entity has some attributes associated to it and has some methods to perform on the attributes.
- **Classes** - A class is a generalized description of an object. An object is an instance of a class. Class defines all the attributes, which an object can have and methods, which defines the functionality of the object.  
In the solution design, attributes are stored as variables and functionalities are defined by means of methods or procedures.
- **Encapsulation** - In OOD, the attributes (data variables) and methods (operation on the data) are bundled together is called encapsulation. Encapsulation not only bundles important information of an object

together, but also restricts access of the data and methods from the outside world. This is called information hiding.

- **Inheritance** - OOD allows similar classes to stack up in hierarchical manner where the lower or sub-classes can import, implement and re-use allowed variables and methods from their immediate super classes. This property of OOD is known as inheritance. This makes it easier to define specific class and to create generalized classes from specific ones.
- **Polymorphism** - OOD languages provide a mechanism where methods performing similar tasks but vary in arguments, can be assigned same name. This is called polymorphism, which allows a single interface performing tasks for different types. Depending upon how the function is invoked, respective portion of the code gets executed.

## 9. Feasibility analysis

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called feasibility study. This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements.

Different Type of Feasibility Study:-

In the conduct of the feasibility study, the analyst will usually considers even distinct, but inter- related types of feasibility. They are Technical Feasibility Operational Feasibility Economic Feasibility Social Feasibility Management Feasibility Legal Feasibility Time Feasibility

### (1) Technical Feasibility:

This is concerned with specifying equipment and software that will successfully satisfy the user requirement; the technical needs of the system may vary considerably, but might include: The facility to produce outputs in a given time:

- Response time under certain conditions.
- Ability to process a certain volume of transaction at a Particular speed.

- Facility to communicate data to distant location. In examine technical feasibility; configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirement: How many workstations are required, how these units are interconnected so that they could operate and communicate smoothly. What speeds of input and output should be achieved at particular quality of printing. This can be used as a basis for the tender document

## (2) Operational Feasibility:-

It is mainly related to human organization and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organizational structures are distributed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time? Generally project will not be rejected simply because of operational infallibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

## (3) Economical Feasibility:-

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More frequently known as cost/ benefit analysis; the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a change of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle

## .(4) Social Feasibility:

Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

## (5) Management Feasibility:-

It is a determination of whether a proposed project will be acceptable to management .if does not accept a project or gives a negligible support to it; the analyst will tend to view the project as a non-feasible one

.(6) Legal Feasibility:-

Legal feasibility is a determination of whether a proposed project infringes on known Acts Statutes, as well as any pending legislation. Although in some instances the project might appear sound, on closer investigation it may be found to infringe on several legal areas.

(7) Time Feasibility:

Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame. If a project takes too much time it is likely to be rejected.



## 10. Project Plan

11. To complete my project I have to go through study of various languages like MERN STACK for backend, Databases for storing retrieving and manipulating data and for front-end I have to learn html, CSS, bootstrap, java script and React JS. Let's have a look on what I have studied and implement it on my project

12.

## INTRODUCTION TO MERN STACK

### Overview

The MERN stack is a popular JavaScript-based full-stack web development framework. It consists of four main components:

1. **MongoDB:** MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It's known for its scalability, flexibility, and ease of use. MongoDB is used as the backend database in MERN stack applications, allowing developers to store and manage data.
2. **Express.js:** Express.js is a minimalist web application framework for Node.js. It provides a robust set of features for building web applications and APIs. Express.js simplifies the process of handling HTTP requests, routing, middleware integration, and more. It's used as the backend server framework in MERN stack applications.
3. **React.js:** React.js is a JavaScript library developed by Facebook for building user interfaces. It allows developers to create reusable UI components that efficiently update and render in response to data changes. React.js follows a component-based architecture, making it easy to manage complex UIs. In the MERN stack, React.js is used for building the frontend client-side of web applications.
4. **Node.js:** Node.js is a JavaScript runtime environment that allows developers to run JavaScript code on the server-side. It's built on the V8 JavaScript engine and provides an event-driven, non-blocking I/O model that's well-suited for building scalable and high-performance web applications. Node.js is used as the runtime environment for server-side code in MERN stack applications.

The MERN stack leverages the power of JavaScript across the entire development stack, allowing developers to use a single language (JavaScript) for both frontend and backend development. This enables greater code reusability, faster development cycles, and easier maintenance. Additionally, the MERN stack offers flexibility and scalability, making it suitable for building a wide range of web applications, from simple prototypes to large-scale, production-ready systems.

## Evolution of technology

The MERN stack, like many technology stacks, has evolved over time in response to changing trends, requirements, and technological advancements. Here's an overview of the evolution of the MERN stack:

### 1. **MongoDB (2009):**

- MongoDB was initially released in 2009 as an open-source NoSQL database. Its development was influenced by the need for a database solution that could handle large volumes of unstructured or semi-structured data, which traditional relational databases struggled with.
- MongoDB's early versions focused on providing features like horizontal scalability, high availability, and flexibility in data modeling. Over time, it has evolved to include features like aggregation pipelines, indexing improvements, and support for various data types.

### 2. **Express.js (2010):**

- Express.js, a minimalist web application framework for Node.js, was released in 2010. It was built to provide a simple and unopinionated framework for building web servers and APIs using Node.js.
- In its early versions, Express.js focused on features like routing, middleware support, and HTTP request handling. As Node.js gained popularity, Express.js became the de facto standard for building backend servers with Node.js due to its simplicity and flexibility.

### 3. **React.js (2013):**

- React.js was developed by Facebook and released in 2013. It was designed to address the challenges of building large-scale, dynamic user interfaces by introducing a component-based architecture and a virtual DOM.

- React.js initially gained traction among front-end developers due to its performance optimizations, declarative syntax, and reusable component model. It quickly became one of the most popular JavaScript libraries for building user interfaces.

#### 4. **Node.js (2009):**

- Node.js, initially released in 2009, is a JavaScript runtime built on Chrome's V8 JavaScript engine. It introduced the concept of server-side JavaScript, allowing developers to use JavaScript for both frontend and backend development.
- Node.js was initially adopted by developers looking to build real-time web applications and APIs. Its non-blocking I/O model and event-driven architecture made it well-suited for handling concurrent connections and asynchronous operations.

#### 5. **MERN Stack (2013 onwards):**

- The MERN stack emerged as a full-stack JavaScript framework, combining MongoDB, Express.js, React.js, and Node.js. Its development was influenced by the growing popularity of JavaScript across the entire development stack.
- Initially, MERN stack applications were used for building single-page applications (SPAs) and real-time web applications. However, over time, the MERN stack has evolved to support a wider range of use cases, including server-rendered applications, progressive web apps (PWAs), and mobile app backends.
- The MERN stack continues to evolve with updates to its individual components, as well as the emergence of new tools, libraries, and best practices in the JavaScript ecosystem.

Overall, the evolution of the MERN stack reflects the broader trends in web development, including the increasing adoption of JavaScript, the rise of NoSQL databases, and the shift towards component-based UI architectures.

### **MERN STACK Features**

The MERN stack offers several features and benefits that make it a popular choice for web development. Here are some key features:

1. **Full-stack JavaScript:** With the MERN stack, developers can use JavaScript for both frontend and backend development. This reduces context switching and allows for better code reuse and maintainability.
2. **Component-based architecture:** React.js, a key component of the MERN stack, utilizes a component-based architecture. This allows developers to build modular and reusable UI components, resulting in cleaner and more maintainable code.
3. **Rich ecosystem:** The MERN stack benefits from a rich ecosystem of libraries, tools, and frameworks that complement its core components. This includes libraries for state management (e.g., Redux), routing (e.g., React Router), and styling (e.g., styled-components).
4. **Scalability:** MongoDB, the database component of the MERN stack, is designed for scalability and can handle large volumes of data and high traffic loads. Combined with the asynchronous and non-blocking nature of Node.js, MERN stack applications can scale easily as the demand grows.
5. **Flexibility:** The MERN stack is highly flexible and can be used to build a wide range of applications, including single-page applications (SPAs), progressive web apps (PWAs), e-commerce platforms, and more. Developers have the freedom to choose the tools and technologies that best fit their project requirements.
6. **Real-time capabilities:** Node.js, the backend component of the MERN stack, is well-suited for building real-time web applications thanks to its event-driven architecture and support for WebSockets. This makes it ideal for applications that require live updates, such as chat apps, collaboration tools, and real-time analytics dashboards.
7. **Community support:** The MERN stack has a large and active community of developers, which means there are plenty of resources, tutorials, and open-source projects available to help developers get started and troubleshoot issues.
8. **Rapid development:** The MERN stack streamlines the development process by providing a unified set of technologies and tools. Developers can leverage pre-built components, libraries, and boilerplate templates to speed up development and focus on building features that add value to their application.

Overall, the MERN stack offers a powerful and flexible framework for building modern web applications, with features that cater to the needs of developers, businesses, and end-users alike.

AMIT KUMAR

## *Introduction to MERN STACK*

The MERN stack is a popular full-stack web development framework that enables developers to build dynamic and interactive web applications using JavaScript-based technologies. The acronym "MERN" stands for MongoDB, Express.js, React.js, and Node.js, which are the four main components of this stack.

Here's a brief introduction to each component:

1. **MongoDB:** MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It is known for its scalability, flexibility, and ease of use. MongoDB is used as the backend database in MERN stack applications, allowing developers to store and manage data in a document-oriented database.
2. **Express.js:** Express.js is a minimalist web application framework for Node.js. It provides a robust set of features for building web servers and APIs using JavaScript. Express.js simplifies the process of handling HTTP requests, routing, middleware integration, and more. It serves as the backend server framework in MERN stack applications.
3. **React.js:** React.js is a JavaScript library developed by Facebook for building user interfaces. It allows developers to create reusable UI components that efficiently update and render in response to data changes. React.js follows a component-based architecture, making it easy to manage complex UIs. In the MERN stack, React.js is used for building the frontend client-side of web applications.
4. **Node.js:** Node.js is a JavaScript runtime environment that allows developers to run JavaScript code on the server-side. It is built on the V8 JavaScript engine and provides an event-driven, non-blocking I/O model that is well-suited for building scalable and high-performance web applications. Node.js is used as the runtime environment for server-side code in MERN stack applications.

The MERN stack leverages the power of JavaScript across the entire development stack, enabling developers to use a single language (JavaScript) for both frontend and backend development. This simplifies the development process, improves code maintainability, and accelerates the development cycle.

Overall, the MERN stack offers a modern and efficient framework for building web applications, with a strong emphasis on flexibility, scalability, and developer productivity.

### Features of MERN Stack

**Rapid Development:** The MERN (MongoDB, Express.js, React.js, Node.js) Stack is renowned for its ability to facilitate rapid development of web applications. With a unified JavaScript-based technology stack, developers can swiftly build and deploy applications, streamlining the development process.

**Secure:** Security is a top priority within the MERN Stack ecosystem. With proper implementation, developers can mitigate common security vulnerabilities such as cross-site scripting (XSS), cross-site request forgery (CSRF), and injection attacks. Frameworks like Express.js offer robust middleware for handling authentication and authorization, ensuring secure user management.

**Scalable:** MERN Stack applications are inherently scalable, thanks to the scalability of Node.js and the flexibility of MongoDB. Node.js allows for asynchronous, non-blocking I/O operations, making it efficient for handling high levels of concurrent requests. MongoDB's document-oriented data model provides horizontal scalability, enabling applications to seamlessly accommodate growing user bases and data volumes.

**Fully Loaded:** The MERN Stack comes equipped with a plethora of tools, libraries, and modules to expedite web development. Express.js simplifies server-side development with its middleware architecture, while React.js offers a rich set of components for building interactive user interfaces. Additionally, Node.js provides a robust runtime environment for executing JavaScript code on the server-side, facilitating full-stack development within a single language.

**Versatile:** The MERN Stack's versatility enables developers to create a wide range of applications across various domains. Whether it's e-commerce platforms, social networking sites, or real-time data visualization tools, the MERN Stack provides the flexibility and scalability to tackle diverse project requirements.

**Open Source:** The MERN Stack is built on open-source technologies, making it freely accessible to developers worldwide. MongoDB, Express.js, React.js, and Node.js are all open-source projects with active developer communities, fostering collaboration and innovation within the ecosystem. This open-source ethos reduces barriers to entry and promotes a culture of knowledge sharing and continuous improvement.

**Vast and Supported Community:** The MERN Stack benefits from a vast and supportive community of developers, enthusiasts, and contributors. From online forums and documentation to meetups and conferences, there are numerous avenues for developers to seek assistance, share insights, and stay updated on the latest advancements in MERN Stack development. This vibrant community ecosystem fosters learning, collaboration, and growth, ensuring that developers have access to resources and support throughout their MERN Stack journey.



## *Introduction to NoSQL databases*

NoSQL, or "Not Only SQL," is a term used to describe databases that are designed to handle large volumes of unstructured, semi-structured, or structured data. Unlike traditional relational databases, which store data in tables with predefined schemas, NoSQL databases use flexible data models that can adapt to changing data requirements. Here's a brief introduction to NoSQL databases:

1. **Flexible Data Models:** NoSQL databases allow for flexible data modeling, enabling developers to store data in various formats such as key-value pairs, documents, graphs, or column-oriented tables. This flexibility makes NoSQL databases well-suited for handling diverse and evolving data types.
2. **Scalability:** NoSQL databases are designed for horizontal scalability, meaning they can easily scale out across multiple servers to handle increasing data volumes and traffic loads. This makes them ideal for applications with high scalability requirements, such as social media platforms, e-commerce websites, and real-time analytics systems.
3. **High Performance:** NoSQL databases are optimized for high performance and low latency, making them suitable for use cases that require fast data retrieval and processing. Many NoSQL databases utilize distributed architectures and in-memory caching to achieve high performance levels.
4. **Schema-less Design:** Unlike relational databases, which enforce rigid schemas upfront, NoSQL databases typically have a schema-less or schema-flexible design. This means developers can store data without predefined schemas, allowing for greater agility and flexibility in application development.
5. **CAP Theorem:** NoSQL databases are often designed with consideration for the CAP theorem, which states that it's impossible for a distributed data store to simultaneously provide consistency, availability, and partition tolerance. NoSQL databases typically prioritize either consistency and availability (CA stores) or availability and partition tolerance (AP stores), depending on the specific use case requirements.
6. **Types of NoSQL Databases:** There are several types of NoSQL databases, each optimized for different use cases:
  - **Document Stores:** Store data in flexible, JSON-like documents (e.g., MongoDB, Couchbase).
  - **Key-Value Stores:** Store data as key-value pairs (e.g., Redis, Amazon DynamoDB).

- **Column-Family Stores:** Store data in columns rather than rows (e.g., Apache Cassandra, HBase).
- **Graph Databases:** Store data in graph structures (e.g., Neo4j, Amazon Neptune).

Overall, NoSQL databases offer a flexible, scalable, and high-performance alternative to traditional relational databases, making them well-suited for modern applications that require efficient handling of diverse and large-scale data sets.

## *NOSQL Features/ Why to use NOSQL*

NoSQL databases offer a range of features that make them attractive for certain use cases. Here are some key features and reasons why you might choose to use a NoSQL database:

1. **Schema Flexibility:** NoSQL databases typically have flexible schemas or no schemas at all, allowing you to store data in a variety of formats without requiring a predefined schema. This flexibility is beneficial for handling semi-structured or unstructured data, as well as accommodating evolving data requirements.
2. **Scalability:** NoSQL databases are designed for horizontal scalability, meaning they can easily scale out across multiple servers to handle increasing data volumes and traffic loads. This makes them well-suited for applications that require high scalability, such as social media platforms, e-commerce websites, and real-time analytics systems.
3. **High Performance:** Many NoSQL databases are optimized for high performance and low latency, enabling fast data retrieval and processing. They often use distributed architectures, in-memory caching, and other optimization techniques to achieve high performance levels.
4. **Support for Large Data Sets:** NoSQL databases excel at handling large volumes of data, making them suitable for big data applications and real-time analytics. They can efficiently store and process massive data sets without sacrificing performance or reliability.
5. **Flexible Data Models:** NoSQL databases support various data models, including document-oriented, key-value, column-family, and graph databases. This flexibility allows you to choose the most appropriate data model for your application's specific requirements.

6. **CAP Theorem Trade-offs:** NoSQL databases are designed with consideration for the CAP theorem, which states that it's impossible for a distributed data store to simultaneously provide consistency, availability, and partition tolerance. NoSQL databases typically prioritize either consistency and availability (CA stores) or availability and partition tolerance (AP stores), depending on the specific use case requirements.
  7. **High Availability and Fault Tolerance:** NoSQL databases are often designed to provide high availability and fault tolerance, ensuring that your data remains accessible even in the event of hardware failures or network partitions. They use replication, sharding, and other techniques to distribute data across multiple nodes and minimize the risk of data loss or downtime.
  8. **Support for Modern Application Architectures:** NoSQL databases are well-suited for modern application architectures, such as microservices, serverless computing, and containerized deployments. They can easily integrate with cloud services, containers, and other infrastructure components commonly used in modern application development.
- Overall, NoSQL databases offer a flexible, scalable, and high-performance alternative to traditional relational databases, making them ideal for modern applications that require efficient handling of diverse and large-scale data sets.

## *Introduction to HTML*

- HTML stands for Hypertext Markup Language.
- HTML is used to create web pages and web applications.
- HTML is widely used language on the web.
- We can create a static website by HTML only.
- Technically, HTML is a Markup language rather than a programming language.

## *Introduction to CSS*

The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.

- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

### *Introduction to Bootstrap*

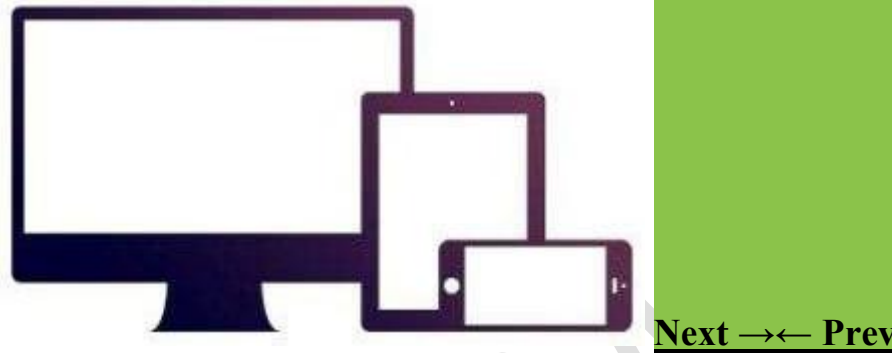
Bootstrap is the popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.

Our Bootstrap tutorial includes all topics of Bootstrap such as jumbotron, table, button, grid, form, image, alert, wells, container, carousel, panels, glyphicon, badges, labels, progress bar, pagination, pager, list group, dropdown, collapse, tabs, pills, navbar, inputs, modals, tooltip, popover and scrollspy.

### *What is Bootstrap*

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.

- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.
- It facilitates you to create responsive designs.



### *What is Bootstrap*

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
  - It is absolutely free to download and use.
  - It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
  - It can also use JavaScript plug-ins.
  - It facilitates you to create responsive designs.



---

### History of Bootstrap

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub.

---

## Why use Bootstrap

Following are the main advantage of Bootstrap:

- It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
    - It facilitates users to develop a responsive website.
  - It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.
- 

## What is a responsive website

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

---

## What Bootstrap package contains

**Scaffolding:** Bootstrap provides a basic structure with Grid System, link styles, and background.

**CSS:** Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

**Components:** Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.

**JavaScript Plugins:** Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.

*Customize: Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.*

## *INTRODUCTION OF REACTJS*

React.js, often referred to simply as React, is an open-source JavaScript library developed and maintained by Facebook. It is widely used for building user interfaces (UIs) for web applications, particularly single-page applications (SPAs) where dynamic and interactive content is key.

Here's an introduction to React.js:

13. **Component-Based Architecture:** React.js follows a component-based architecture, which means UIs are built using reusable components. Components are self-contained pieces of code that encapsulate a part of the user interface. This modular approach makes it easier to manage complex UIs, improve code reusability, and maintain consistency across different parts of an application.
14. **Virtual DOM (Document Object Model):** React.js utilizes a virtual DOM, a lightweight representation of the actual DOM in memory. When there are changes to a component's state or props, React compares the virtual DOM with the real DOM and updates only the parts that have changed. This minimizes the number of DOM manipulations and improves performance, resulting in faster rendering and better user experience.
15. **Declarative Syntax:** React.js promotes a declarative programming style, where developers describe how the UI should look and behave based on the current state, rather than imperatively specifying each step of the UI update process. This makes code easier to understand, maintain, and debug.
16. **JSX (JavaScript XML):** React.js uses JSX, a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript files. JSX makes it easier to define UI components and their structure, as well as incorporate JavaScript expressions and logic directly into the UI code.
17. **One-Way Data Flow:** React.js follows a unidirectional data flow model, where data flows downwards from parent components to child components via props. This helps maintain data consistency and predictability, as changes to data in parent components automatically propagate to child components.
18. **React Hooks:** React introduced Hooks in version 16.8, which are functions that allow developers to use state and other React features without writing class components. Hooks provide a simpler and more concise way to manage component state, side effects, and other React features.

19. **Rich Ecosystem:** React.js has a rich ecosystem of libraries, tools, and frameworks that complement its core features. This includes libraries for state management (e.g., Redux, MobX), routing (e.g., React Router), styling (e.g., styled-components, CSS Modules), and more. These libraries help extend React's capabilities and improve developer productivity.

20. **Community and Support:** React.js has a large and active community of developers, which means there are plenty of resources, tutorials, and open-source projects available to help developers learn React and solve common problems. The React community also contributes to the ongoing development and improvement of the library.

Overall, React.js is a powerful and flexible JavaScript library for building modern web applications with dynamic and interactive user interfaces. Its component-based architecture, virtual DOM, declarative syntax, and rich ecosystem make it a popular choice among developers for building scalable and maintainable UIs.

1. **React Hooks:** React introduced Hooks in version 16.8, which are functions that allow developers to use state and other React features without writing class components. Hooks provide a simpler and more concise way to manage component state, side effects, and other React features.

2. **Rich Ecosystem:** React has a vast ecosystem of libraries, tools, and frameworks that complement its core features. This includes libraries for state management (e.g., Redux, MobX), routing (e.g., React Router), styling (e.g., styled-components, CSS Modules), and more. These libraries help extend React's capabilities and improve developer productivity.

Overall, React.js is a powerful and flexible JavaScript library for building modern web applications with dynamic and interactive user interfaces. Its component-based architecture, virtual DOM, declarative syntax, and rich ecosystem make it a popular choice among developers for creating scalable, maintainable, and performant UIs.

---



*CODE Snap Shots*

JS server.js X

JS server.js &gt; ...

```
1  const express = require('express');
2  const cors = require('cors');
3  const connectDB = require('./db');
4  const AddroomRoutes = require('./routes/AddroomRoutes');
5  const OwnerRoutes = require('./routes/Hotel-OwnerRoutes');
6  const UserRoutes = require('./routes/UserRoutes');
7  const AdminRoutes = require('./routes/AdminRoutes');
8  const BookRoomRoutes = require('./routes/BookRoomRoutes');
9  const app = express();
10
11  // Connect to MongoDB
12  connectDB();
13
14  // Middleware
15  app.use(express.json());
16  app.use(cors());
17
18  // Routes
19  app.use('/api/Admin', AdminRoutes);
20  app.use('/api/rooms', AddroomRoutes);
21  app.use('/api/Owners', OwnerRoutes);
22  app.use('/api/users', UserRoutes);
23  app.use('/api/bookroom', BookRoomRoutes);
```

```

BookNow.jsx X BookRoom.jsx
src > pages > BookNow.jsx > Book
1  import { useEffect, useState } from "react";
2  import { useSearchParams } from "react-router-dom";
3  import { useNavigate } from 'react-router-dom';
4  import { ToastContainer, toast } from "react-toastify";
5  import 'react-toastify/dist/ReactToastify.css';
6  import './css/book.css';
7
8  const Book = () => {
9      const navigate = useNavigate();
10     const [searchParams] = useSearchParams();
11     const roomId = searchParams.get("roomId") || "";
12     const [startDate, setStartDate] = useState("");
13     const [endDate, setEndDate] = useState("");
14     const [availability, setAvailability] = useState(null);
15
16     useEffect(() => {
17         if (!roomId) {
18             toast.error("Room ID is missing.");
19         }
20     }, [roomId]);
21

```

```

JS AddRoomRoutes.js X
routes > JS AddRoomRoutes.js > ...
1  const express = require('express');
2  const mongoose = require('mongoose');
3  const cors = require('cors');
4  const app = express();
5  const Room = require ('../models/AddRoomSchema');
6
7  app.use(express.json());
8  app.use(cors());
9
10 // Get all rooms
11 app.get('/', async (req, res) => {
12     try {
13         const rooms = await Room.find();
14         res.status(200).json(rooms);
15     } catch (error) {
16         res.status(500).json({ message: 'Failed to fetch rooms.', error });
17     }
18 });
19

```

```

JS server.js  X
JS server.js > ...
1  const express = require('express');
2  const cors = require('cors');
3  const connectDB = require('./db');
4  const AddroomRoutes = require('./routes/AddroomRoutes');
5  const OwnerRoutes = require('./routes/Hotel-OwnerRoutes');
6  const UserRoutes = require('./routes/UserRoutes');
7  const AdminRoutes = require('./routes/AdminRoutes');
8  const BookRoomRoutes = require('./routes/BookRoomRoutes');
9  const app = express();
10
11  // Connect to MongoDB
12  connectDB();
13
14  // Middleware
15  app.use(express.json());
16  app.use(cors());
17
18  // Routes
19  app.use('/api/Admin', AdminRoutes);
20  app.use('/api/rooms', AddroomRoutes);
21  app.use('/api/Owners', OwnerRoutes);
22  app.use('/api/users', UserRoutes);
23  app.use('/api/bookroom', BookRoomRoutes);

```

```

JS db.js  X
JS db.js > ...
1  const mongoose = require('mongoose');
2  const dbURL= 'mongodb://localhost:27017/Amit-Hotel-Project';
3
4  const connectDB = async () => {
5    try {
6      await mongoose.connect(dbURL, {
7        useNewUrlParser: true,
8        useUnifiedTopology: true,
9      });
10     console.log('MongoDB connected successfully');
11   } catch (error) {
12     console.error('Error connecting to MongoDB:', error);
13     process.exit(1);
14   }
15 };
16
17 module.exports = connectDB;

```

## 21. TEST PLAN

### 1. Functionality Testing a Website of

**Functionality Testing of a Website** is a process that includes several testing parameters like user interface, APIs, database testing, security testing, client and server testing and basic website functionalities. Functional testing is very convenient and it allows users to perform both manual and automated testing. It is performed to test the functionalities of each feature on the website.

Web based Testing Activities includes:

**Test all links** in your webpages are working correctly and make sure there are no broken links. Links to be checked will include -

- Outgoing links
- Internal links
- Anchor Links
- MailTo Links

**Test Forms** are working as expected. This will include-

- Scripting checks on the form are working as expected. For example- if a user does not fill a mandatory field in a form an error message is shown.
- Check default values are being populated
- Once submitted, the data in the forms is submitted to a live database or is linked to a working email address
- Forms are optimally formatted for better readability

**Test Cookies** are working as expected. Cookies are small files used by websites to primarily remember active user sessions so you do not need to log in every time you visit a website. Cookie Testing will include

- Testing cookies (sessions) are deleted either when cache is cleared or when they reach their expiry.
- Delete cookies (sessions) and test that login credentials are asked for when you next visit the site.

**Test HTML and CSS** to ensure that search engines can crawl your site easily. This will include

- Checking for Syntax Errors
- Readable Color Schemas

- **Standard Compliance.** Ensure standards such W3C, OASIS, IETF, ISO, ECMA, or WS-I are followed.

**Test business workflow-** This will include

- Testing your end - to - end workflow/ business scenarios which takes the user through a series of webpages to complete.
- Test negative scenarios as well, such that when a user executes an unexpected step, appropriate error message or help is shown in your web application.

**Tools that can be used:** QTP , IBM Rational , Selenium

## 2. Usability testing:

Usability Testing has now become a vital part of any web based project. It can be **carried out by testers** like you **or a small focus group** similar to the target audience of the web application.

**Test the site Navigation:**

- Menus, buttons or Links to different pages on your site should be easily visible and consistent on all webpages

**Test the Content:**

- Content should be legible with no spelling or grammatical errors.
- Images if present should contain an "alt" text

**Tools that can be used:** Chalkmark, Clicktale, Clixpy and Feedback Army

## 3.Interface Testing:

Three areas to be tested here are - Application, Web and Database Server

- **Application:** Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.
- **Web Server:** Test Web server is handling all application requests without any service denial.
- **Database Server:** Make sure queries sent to the database give expected results.

**Test system response when connection between the three layers (Application, Web and Database) cannot be established** and appropriate message is shown to the end user.

**Tools that can be used:** AlertFox, Ranorex

#### 4. Database Testing:

Database is one critical component of your web application and stress must be laid to test it thoroughly. Testing activities will include-

- Test if any errors are shown while executing queries
- Data Integrity is maintained while creating, updating or deleting data in database.
- Check response time of queries and fine tune them if necessary.
- Test data retrieved from your database is shown accurately in your web application

**Tools that can be used:** QTP, Selenium

#### 5. Compatibility testing.

Compatibility tests ensures that your web application displays correctly across different devices. This would include-

**Browser Compatibility Test:** Same website in different browsers will display differently. You need to test if your web application is being displayed correctly across browsers, JavaScript, AJAX and authentication is working fine. You may also check for Mobile Browser Compatibility. The rendering of web elements like buttons, text fields etc. changes with change in **Operating System**. Make sure your website works fine for various combination of Operating systems such as Windows, Linux, Mac and Browsers such as Firefox, Internet Explorer, Safari etc.

#### 6. Performance Testing:

This will ensure your site works under all loads. Software Testing activities will include but not limited to -

- Website application response times at different connection speeds
- Load test your web application to determine its behavior under normal and peak loads
- Stress test your web site to determine its break point when pushed to beyond normal loads at peak time.

- Test if a crash occurs due to peak load, how does the site recover from such an event
- Make sure optimization techniques like gzip compression, browser and server side cache enabled to reduce load times

#### 7. Security testing:

Security Testing is vital for e-commerce website that store sensitive customer information like credit cards. Testing Activities will include-

- Test unauthorized access to secure pages should not be permitted
- Restricted files should not be downloadable without appropriate access
- Check sessions are automatically killed after prolonged user inactivity
- On use of SSL certificates, website should re-direct to encrypted SSL pages.

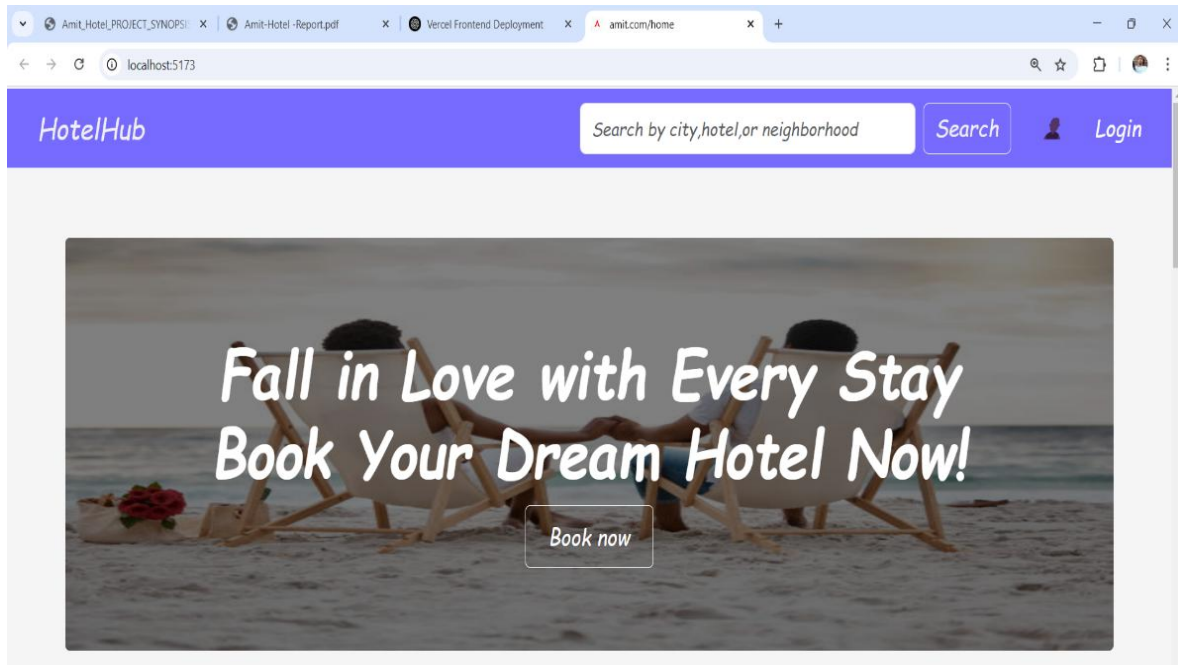
#### 8. Crowd Testing:

You will select a large number of people (crowd) to execute tests which otherwise would have been executed a select group of people in the company. Crowdsourced testing is an interesting and upcoming concept and helps unravel many a unnoticed defects.

## Project Legacy

### Current Snapshots:

#### Landing page

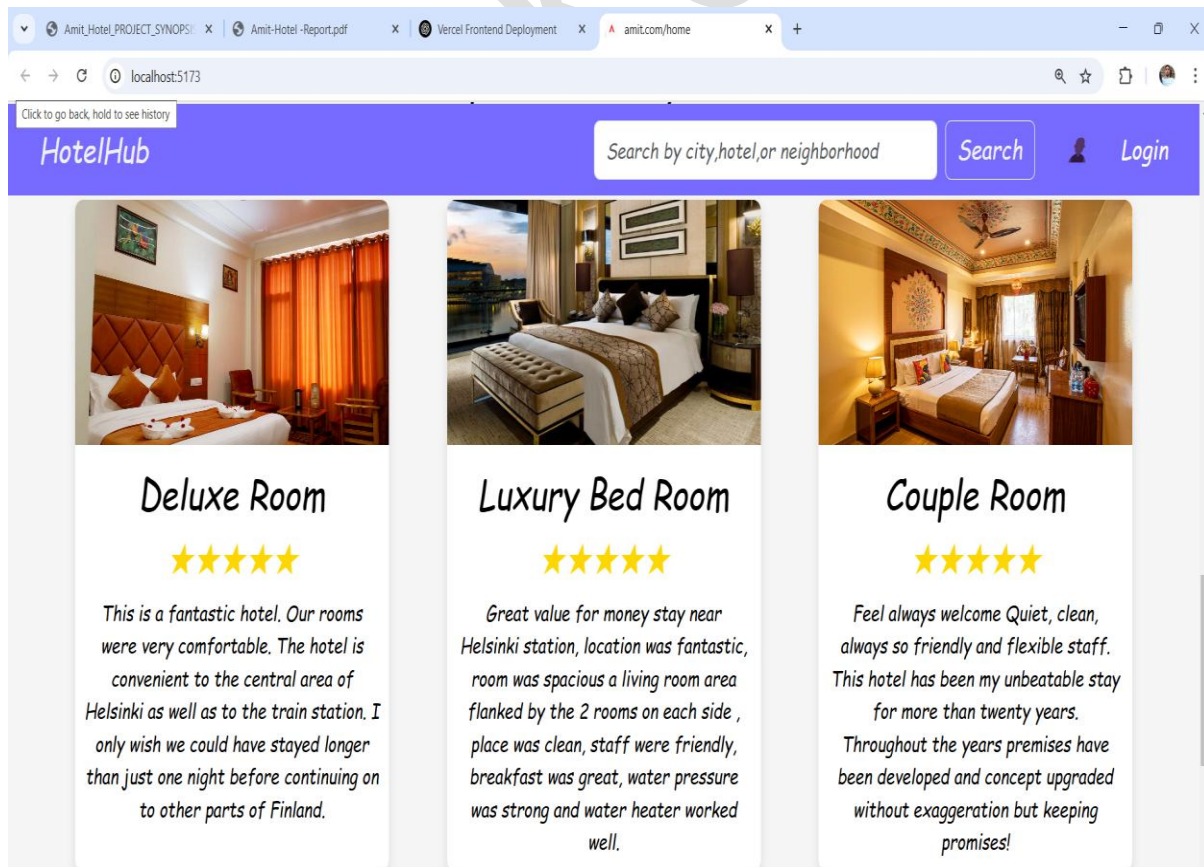
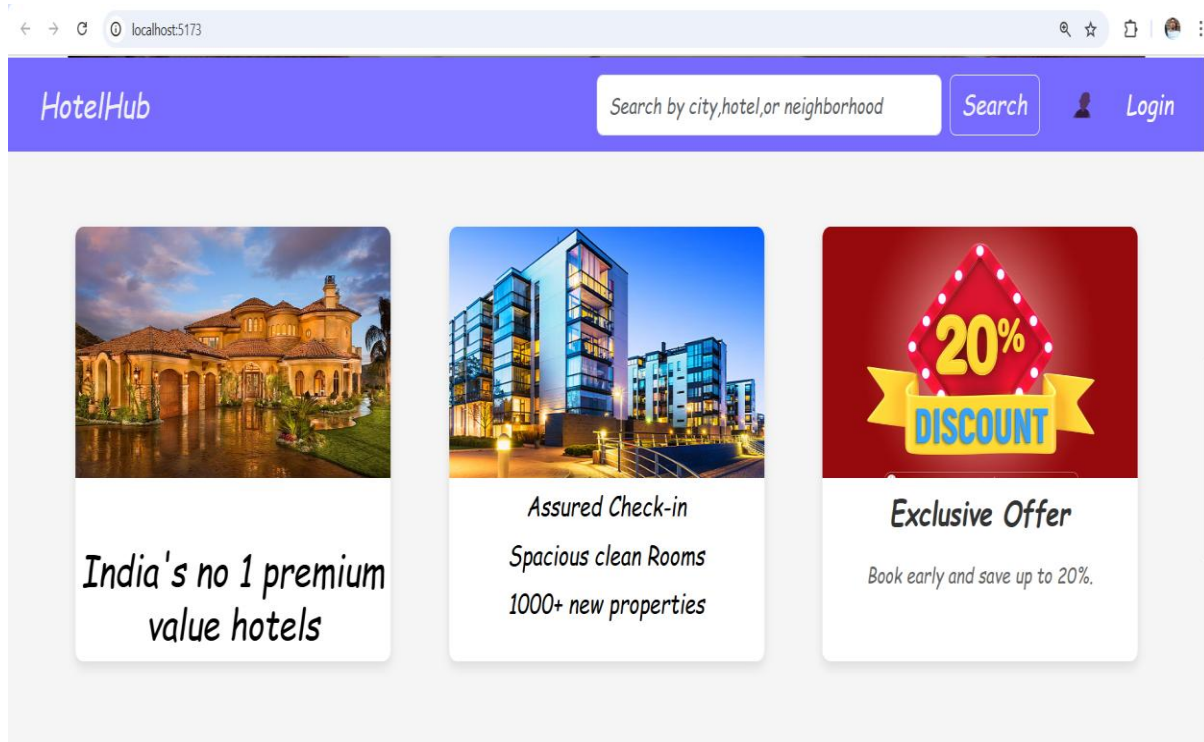


#### Signup page

A screenshot of a 'Sign-Up' form. The title 'Sign-Up' is centered at the top in a large, italicized font. Below the title are three input fields with placeholder text: 'Enter Your UserName', 'Enter Your Email', and 'Enter Your Password'. Below these fields is a solid blue button with the text 'Sign-Up' in white. Underneath the button is the text 'Already have an account?' in an italicized font. At the bottom is another solid blue button with the text 'Log-In' in white.



Previous or



localhost:5173/room


Click to go back, hold to see history

HotelHub


Search by city,hotel,or neighborhood

Search

Login



**Vishal Hotel** ★★★★★  
Room-Type : Deluxe Room  
Max Guests: 2  
City : Haryana  
Area : Ambala  
Price: 1999 ₹ per night  
Includes taxes and charges  
[SEE AVAILABILITY](#)



**Hotel Sky-Blue** ★★★★★  
Room-Type : Luxury Bed Room  
Max Guests: 2  
Price: 1999 ₹ per night  
Includes taxes and charges

## Room Booking

Room ID: **67a59e8f090d6a3c122518bd**

**Start Date:**

dd-mm-yyyy



**End Date:**

dd-mm-yyyy



[Check Availability](#)

## Book Your Room





localhost:5173/about

HotelHub




Login

## About Us

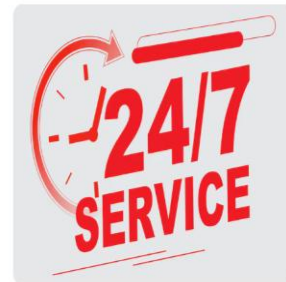
Welcome to India's No.1 Premium Value Hotels! We provide guests with the perfect blend of comfort, style, and affordability. Our hotel network is built to cater to travelers seeking a luxurious stay without compromising on quality.



Premium Stays



Delicious Meals



24/7 Customer Service

## Product management system

### *Benefits of E-commerce website*

1. **Real-Time Communication:** MERN stack, especially with technologies like Socket.IO, enables real-time communication between users.

3. **Single Page Application (SPA):** With React.js on the frontend, you can create a highly responsive and interactive user interface. SPAs provide a smoother user experience by loading only the necessary components instead of refreshing the entire page.

4. **Cross-Platform Compatibility:** MERN stack allows you to develop applications that are compatible with multiple platforms, including web browsers and mobile devices. React Native, a framework based on React.js, can be used to develop native mobile apps with the same codebase.

5. **Flexibility and Modularity:** Each component of the MERN stack is modular and can be replaced with alternative technologies if needed. This provides developers with the flexibility to choose the best tools for their specific requirements.

6. **Rich Ecosystem:** The MERN stack has a large and active community, which means there are plenty of resources, tutorials, and third-party libraries available to help developers build robust and feature-rich chat applications efficiently.

7. **Security:** By implementing proper authentication and authorization mechanisms using tools like JWT (JSON Web Tokens) and encryption techniques, you can ensure the security of your chat application, protecting user data and preventing unauthorized access.

8. **Rapid Development:** MERN stack promotes rapid development through its use of JavaScript across the entire stack. Developers can leverage libraries, frameworks, and tools to speed up the development process and deliver the application to market faster.

9. **SEO-Friendly:** Despite being a single-page application, MERN stack applications can still be made SEO-friendly using server-side rendering techniques, ensuring that your chat application is discoverable by search engines.

- 9. Cost-Effectiveness:** Since MERN stack technologies are open-source and readily available, building a chat application using these technologies can be cost-effective, especially for startups and small businesses.

## REFERENCES AND BIBLIOGRAPHY

- [Mongo DB](#)
- [React](#)
- [Node.js](#)