PURNEA COLLEGE OF ENGINEERING , PURNEA

AN INTERNSHIP REPORT ON

Full Stack Web Development

UNDERTAKEN

INVIGO INFOTECH

Submitted in partial fulfillment of the requirements for the award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

(19 May 2025 to 28 June 2025)

Submitted By :

Name : **Pawan Kumar Malakar**

Branch :**CSE**

Reg. :**23105111005**

Semester : **3th**

Session : **2023-2027**

Submitted To :

.....

Prof. Mr. **Ajit Kumar Gupta**

(H.O.D of CSE)

.....

Prof.

(Internal Examiner)

.....

(External Examiner)

INTERNSHIP REPORT INDEX

Topic	Page Number
Cover Page	1
Declaration	2
Acknowledgment	3
Internship Certificate	4-5
Abstract	6
Weekly Report - Week 1 (HTML)	7
Weekly Report - Week 2 (CSS)	8
Weekly Report - Week 3 (JavaScript)	9
Weekly Report - Week 4 (jQuery)	9
Weekly Report - Week 5 (Node.js, Express, MongoDB)	10
Weekly Report - Week 6 (Database, Session, Project)	11
Technology Description - Web Development	12
Technology - HTML	13-16
Technology - CSS	17-19
Technology - JavaScript	20-23
Technology - Bootstrap	24
Technology - DBMS	24
Technology - MySQL	25
Technology - React.js	25-26
Technology - Node.js	26-27
Technology - Express.js	27-28
Technology - MongoDB	28-29
Technology - React.js (Continued)	29-30
Project Description - ApnaRoom.com	31-34
Frontend & Backend Details	35-36
Conclusion	37
References	38

DECLARATION

I **PAWAN KUMAR MALAKAR** hereby declare that I have undertaken 2 Weeks online as well as 4 Weeks ofline internship training program at

" **INVIGO INFOTECH**" During a period from 19th May to 31th May, 2025

online and from 1st June to 28th june, 2025 ofline in partial fulfilment for the award of the **Bachelor of Technology in Computer Science & Engineering** from Purnea college of Engineering, Purnea, by Bihar

Engineering University, Patna. The work which is being presented in the internship report submitted to Department of Computer Science & Engineering from Purnea College of Engineering, Purnea is an authentic record of training work.

I also declare that neither this internship nor any part of this internship has been submitted elsewhere for award of any degree.

PAWAN KUMAR MALAKAR

REG. NO: 23105111005

DATE : _____

ACKNOWLEDGEMENT

I am grateful to (INVIGO INFOTECH) trainings for giving me opportunity to carry out the project work in the area of console-based portal designing during my training. I would like to also thank my institute, **PURNEA COLLEGE OF ENGINEERING, PURNEA** for giving
permission and necessary administrative support to take up the training with INVIGO INFOTECH trainings.

Deepest thanks to our Trainers **(INVIGO INFOTECH)** team for their guidance, monitoring, constant encouragement and correcting various assignments of ours with attention and care. They have taken pain to go through the project and training sessions and make necessary corrections as when needed and I am very grateful for that.

INTERNSHIP CERTIFICATE

CIN:- U85490BR2024PTC069312

AICTE Approved

INVIGO INFOTECH

Certificate of Internship

This is Certify that **PAWAN KUMAR MALAKAR** of **Darbhanga College of Engineering, Darbhanga** has successfully completed a **4 Weeks Offline Internship Training** on "**Full Stack Web Development**" during **(01 Jun 2025 - 28 Jun 2025)** in **Invigo Infotech**. During this training, we found candidate sincere, hardworking, technically sound & result oriented. We wish all the best for future endeavors.

Issued on 28 Jun, 2025

Priyanshu Kumar

Founder (Invigo Infotech)

Certificate no: INVI25DB13

Certificate Verification: https://www.invigoinfotech.in/certificate-verification

Training Certificate Link :-

https://drive.google.com/file/d/1Bi90btDOFTluLBzjsYT6buYzXXf84ZAG/view?usp=sharing

ABSTRACT

1. **Name of The Student:-** PAWAN KUMAR MALAKAR
2. **Reg. Number:-** 23105111005
3. **Training Status : -** Yes
   - **Industry Name: -** INVIGO INFOTECH.PVT.LTD
   - **Industry Address: -** Prabhat Colony, Purnea 854301, Bihar
4. **Training Duration:-** May-June
5. **Training Days:-** 2 week online and 4 week ofline
6. **Training certificate:-** Yes
7. **Technology Name:-** HTML, CSS, Javascript , Bootstrap, Node, Express, MongoDB ,React
8. **Project:-** https://github.com/manishdev26/Apna-Room

WEEKLY REPORT :-

WEEK: 1

Description of activity, task, duty or responsibility	Performed with team	Performed Alone	Time Spent
Introduction to HTML		Yes	1 lecture
Basic Tags and HTML Lists		Yes	1 lecture
HTML Tables and Attributes		Yes	1 lecture
introduction to Forms and Form Validation		Yes	1 lecture

WEEK: 2

Description of activity, task, duty or responsibility	Performed with team	Performed Alone	Time Spent
CSS Basics		Yes	1 lecture
Floating		Yes	1 lecture
Styling Texts		Yes	1 lecture
Text and Fonts using Google Fonts		Yes	1 lecture

WEEK: 3

Description of activity,task,duty or responsibility	**Performed with team**	**Performed Alone**	**Time Spent**
Introduction To Javascript Section		Yes	1 lecture
Manipulating Styles With Javascript		Yes	1 lecture
If else and Loops		Yes	1 lecture
Functions and External JS		Yes	1 lecture

WEEK: 4

Description of activity,task,duty or responsibility	Performed with team	Performed Alone	Time Spent
Introduction To jQuery Section		Yes	1 lecture
Changing Website Content		Yes	1 lecture
Animating Content		Yes	1 lecture
Regular Expressions and Form Validation		Yes	1 lecture

WEEK: 5

Description of activity,task,duty or responsibility	**Performed with team**	**Performed Alone**	**Time Spent**
Introduction To Node.js,Express		Yes	1 lecture
GET and POST variables		Yes	1 lecture
Loops in Express		Yes	1 lecture
Introduction to MongoDB		Yes	1 lecture

WEEK: 6

Description of activity,task,duty or responsibility	Performed with team	Performed Alone	Time Spent
Connecting and Retrieving Data from A Database		Yes	1 lecture
Inserting And Updating Data		Yes	1 lecture
Session Variables and Cookies		Yes	1 lecture
Project Website - ApnaRoom.com		Yes	5 lecture

Technology Description:-

WEB DEVELOPMENT :-

Web development is the process of creating and building websites and web applications that you use on the internet. It involves designing what the website looks like (frontend) and making sure it works correctly behind the scenes (backend).

Key Points:

1. **Frontend**: This is what users see and interact with on the website. It includes everything from the layout, colors, and fonts to buttons and forms. It's created using HTML (for structure), CSS (for style), and JavaScript (for interactivity).
2. **Backend**: This is what happens behind the scenes. It handles how the website works, including things like storing data, handling user logins, and processing information. It's built using programming languages like Python, JavaScript (Node.js), Ruby, and PHP, and often involves working with databases.
3. **Full-Stack Development**: Some developers work on both the frontend and backend, and this is called full-stack development.
4. **Responsive Design**: Ensures the website looks good and works well on all devices, like phones, tablets, and computers.
5. **Maintenance**: Once a website is live, it needs regular updates and fixes to keep it running smoothly.

Software Requirements :-

- HTML

- CSS
- JAVASCRIPT
- DBMS
- MongoDB
- MYSQL
- REACT

HTML :

HTML (HyperText Markup Language) is the standard language used to create and design documents on the web. **It's the backbone of every website**, structuring the content and defining the layout

1. Basic Structure

`

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
```

```
        <h1>My First Heading</h1>
        <p>My first paragraph.</p>

    </body>
</html>
```

- **Doctype Declaration**: Every HTML document starts with a <!DOCTYPE html> declaration. This tells the web browser that the document is written in HTML5.
- **HTML Tag (<html>)**: This is the root element of an HTML page. All other elements are nested inside this tag.
- **Head Section (<head>)**: Contains meta-information about the document such as the title (<title>), character set (<meta charset="UTF-8">), and links to external resources like stylesheets or scripts.
- **Body Section (<body>)**: This is where the actual content of the webpage is placed. It includes text, images, links, and other multimedia.

2. HTML Tags and Elements

HTML uses tags to define elements. Tags are enclosed in angle brackets (< >) and often come in pairs: an opening tag and a closing tag (e.g., and for paragraphs).

- **Headings (<h1> to <h6>)**: Define headings on a page. <h1> is the most important, and <h6> is the least important.

- **Paragraphs (<p>)**: Define a block of text as a paragraph.
- **Links (<a href="URL">)**: Create hyperlinks to other pages or resources. The href attribute specifies the destination.
- **Images (<img src="URL" alt="description">)**: Embed images. The src attribute specifies the image source, and alt provides alternative text if the image fails to load.
- **Lists**: There are two types:
  - **Ordered Lists (<ol>)**: Lists items in a specific order. ○ **Unordered Lists (<ul>)**: Lists items without order.
  - **List Items (<li>)**: Defines each item in a list.
- **Tables (<table>)**: Used to display tabular data. ○ **Rows (<tr>)**: Define a row in the table.
  - **Cells (<td>)**: Define a cell in a row.
  - **Headers (<th>)**: Define a header cell, usually bold and centered.
- **Forms (<form>)**: Used to collect user input. Forms can contain various input elements like text fields, checkboxes, radio buttons, and submit buttons.

3. Attributes

HTML tags can have attributes that provide additional information. Attributes are always included in the opening tag and come in keyvalue pairs.

- **Common attributes include:**
  - id: Unique identifier for an element.

- class: Specifies one or more class names for an element. ○ style: Inline CSS to style an element.
- src: Specifies the source of an embedded resource (like an image or script).
- href: Specifies the destination of a link. ○ alt: Provides alternative text for an image.

4. Semantic HTML

Semantic HTML uses tags that clearly describe their meaning. This helps both with accessibility and search engine optimization (SEO).

- **Examples of semantic elements:**
  - <header>: Represents the introductory content or navigational links.
  - <nav>: Defines a section of navigation links.
  - <article>: Represents an independent piece of content, such as a blog post.
  - <section>: Defines a thematic grouping of content, typically with a heading.
  - <footer>: Represents the footer of a document or section, often containing information like the author, copyright, or links to related documents.

Features of HTML

- It is easy to learn and easy to use.
- It is platform-independent.

- Images, videos, and audio can be added to a web page.
- Hypertext can be added to the text.
- It is a markup language.

CSS:

CSS (Cascading Style Sheets) is used to style and lay out web pages — for example, to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

CSS Syntax:-

- **Selector**: The selector targets the HTML element(s) you want to style. It can be a specific element, a group of elements, or elements with a certain class or ID.
- **Declaration Block**: The declaration block contains one or more declarations enclosed in curly braces {}. Each declaration includes a property and a value, separated by a colon.
    - **Property**: This specifies the aspect of the element to be styled (e.g., color, font-size).
    - **Value**: This specifies the setting for the property (e.g., red, 16px).

Type of selectors :-

- **Class Selector**: Selects elements with a specific class. Classes are denoted with a period (.) followed by the class name.

- **ID Selector**: Selects an element with a specific ID. IDs are denoted with a hash symbol (#) followed by the ID name.

- **Universal Selector**: Selects all elements.

- **Descendant Selector**: Selects elements that are descendants of another element.

Box Model:-

The CSS box model is a fundamental concept that dictates how elements are rendered on the page. Every element is treated as a rectangular box, consisting of the following parts:

- **Content**: The actual content inside the element, such as text or images.
- **Padding**: The space between the content and the element's border.
- **Border**: The edge surrounding the padding (if any) and content.
- **Margin**: The space outside the border, separating the element from neighboring elements.

Understanding the box model is crucial for accurately controlling layout and spacing.

Positioning and Layout:-

CSS provides several properties to control the layout and positioning of elements:

- **Static Positioning**: The default positioning method where elements are placed according to the normal document flow.
- **Relative Positioning**: The element is positioned relative to its normal position, allowing you to move it without affecting the flow of surrounding elements.
- **Absolute Positioning**: The element is positioned relative to its closest positioned ancestor (not the initial containing block), and it is removed from the normal document flow.
- **Fixed Positioning**: The element is positioned relative to the browser window, and it remains fixed in place even when the page is scrolled.
- **Flexbox and Grid Layout**: Modern CSS includes powerful layout systems like Flexbox and CSS Grid that offer more flexibility and control over the arrangement of elements.

Responsive Web Design:-

- **Media Queries**: Allow you to apply different styles based on the characteristics of the device, such as screen width.

JavaScript:

Client-side: It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client side are AngularJS, ReactJS, VueJS, and so many others

Server-side: It supplies objects relevant to running JavaScript on a server. For if the server-side extensions allow an application to communicate with a database and provide continuity of information from one invocation to another of the application, or perform f ile manipulations on a server. The useful framework which is the most famous these days is node.js.

Imperative language – We are mostly concerned about how it should be done in this type of language. It simply controls the flow of computation. The procedural programming approach, object, oriented approach comes under this as async-await we are thinking about what is to be done further after the async call.

Declarative programming – In this type of language we are concerned about how it is to be done, basically here logical computation requires. Her main goal is to describe the desired result without direct dictation on how to get it as the arrow function does.

Basic JavaScript Syntax:-

JavaScript code is written using a combination of keywords, operators, variables, and functions. Here are some fundamental elements of JavaScript syntax:

- **Statements**: Instructions that the browser executes. Statements end with a semicolon (;), though it's often optional due to automatic semicolon insertion.

- **Variables**: Containers for storing data values. Variables are declared using var, let, or const.

- **Data Types**: JavaScript has several data types, including:
  - **Primitive Types**: Number, String, Boolean, Undefined, Null, Symbol (ES6), and BigInt (ES11).
  - **Object Types**: Includes Object, Array, Function, Date, RegExp, etc.

- **Operators**: Symbols used to perform operations on variables and values.
  - **Arithmetic Operators**: +, -, *, /, %, etc.
  - **Comparison Operators**: ==, ===, !=, !==, >, <, >=, <= ○ **Logical Operators**: &&, ||, ! ○ **Assignment Operators**: =, +=, -=, *=, /=

Control Structures:-

JavaScript uses control structures to manage the flow of execution in programs:

- **Conditional Statements**: Used to execute code based on conditions.

- **if Statement**: Executes code if a condition is true. ○ **else if and else**: Provide alternative conditions.
- **switch Statement**: Selects one of many code blocks to execute.

- **Loops**: Execute a block of code multiple times.
  - **for Loop**: Iterates over a range of values.
  - **while Loop**: Continues to execute as long as a condition is true.
  - **do...while Loop**: Executes the code block once before checking the condition.

- **Functions:**

  Functions are blocks of code designed to perform specific tasks. They can be defined and invoked as needed.

  - **Function Declaration**: Defines a function that can be called by its name.

  - **Function Expression**: Assigns a function to a variable.

  - **Arrow Functions** (ES6): A shorter syntax for writing functions.

- **Parameters and Arguments**: Parameters are placeholders in function definitions, while arguments are the actual values passed to the function.

- **Objects:**

  Objects are collections of key-value pairs. They are used to group related data and functions.

  - **Creating an Object**: - **Accessing Properties**: - **Adding and Modifying Properties**:

- **Arrays:**
  - Creating an Array -
  Array Methods:
    - **push and pop**: Add or remove elements from the end.
    - **shift and unshift**: Add or remove elements from the beginning.
    - **forEach, map, filter, reduce**: Methods for iterating and manipulating arrays

BOOTSTRAP:

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

DBMS:

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation of the data in the database or the management of the database structure itself. Database Management Systems (DBMSs) are categorized according to their data structures or types.

MYSQL:

MySQL is an open-source relational database management system (RDBMS) widely used for storing, managing, and retrieving structured data. It is known for its performance, reliability, scalability, and ease of use, making it a popular choice for many applications and websites.

MySQL uses Structured Query Language (SQL) to interact with the database. SQL allows developers to perform various operations, such as creating and modifying database structures, inserting and retrieving data, updating records, and executing complex queries for data analysis.

MySQL provides a wide range of data types to accommodate different types of data, including integers, floating-point numbers, strings, dates, timestamps, and more. It also offers support for handling large objects (BLOBs and CLOBs), spatial data, and JSON.

REACT:

ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook. React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more

predictable and easier to debug. A React application is made of multiple components, each responsible for rendering a small, reusable piece of HTML. Components can be nested within other components to allow complex applications to be built out of simple building blocks. A component may also maintain an internal state – for example, a TabList component may store a variable corresponding to the currently open tab.

✅ 6. Node.js

- Node.js is a JavaScript runtime that runs on the server.
- Allows building backend applications using JavaScript.
- Built on Chrome's V8 engine, making it fast and efficient.

🔵 Features:

- Handles HTTP requests
- Reads/writes files

- Connects with databases (MongoDB, MySQL)
- Supports modules (like fs, http, express)

⬭ Example:

js

CopyEdit

```
const http = require("http");

http.createServer((req, res) => {
  res.write("Hello from Node.js!");
  res.end();
}).listen(3000);
```

- Used with npm to manage packages like express, mongoose, dotenv, etc.

✅ 7. Express.js

- Express is a web framework for Node.js to build servers and APIs.
- It simplifies routing, middleware, request/response handling.

⬭ Features:

- Route handling: GET, POST, PUT, DELETE
- Middleware support

- Easy integration with templates and databases

⬭ Example:

```
const express = require("express");
const app = express();

app.get("/", (req, res) => {
  res.send("Hello from Express!");
});

app.listen(3000);
```

- Express makes it easy to create RESTful APIs and full-stack apps.

✅ 8. MongoDB

- MongoDB is a NoSQL database that stores data in JSON-like documents (called BSON).
- Very flexible, no strict schema required.
- Used with Node.js via Mongoose library.

⬭ Example Document:

```
{
```

```
  "name": "John",
  "age": 25,
  "skills": ["JavaScript", "Node.js"]
}
```

⬭ Commands:

- db.collection.insertOne()
- db.collection.find()
- db.collection.updateOne()
- db.collection.deleteOne()
- MongoDB works well for dynamic data, unlike SQL which is strict.

✅ 9. React.js

- React is a JavaScript library for building user interfaces.
- Developed by Facebook. Used to build single-page applications (SPAs).
- Uses a component-based approach.

◆

- JSX syntax: HTML in JavaScript
- Reusable components
- Virtual DOM for fast updates
- State and props for data handling

⬭ Example:

```
function Welcome() {
  return <h1>Hello, React!</h1>;
}
```

- You can use React with hooks, router, and Redux for full apps.
- React is fast, scalable, and widely used in industry.

MY PROJECT

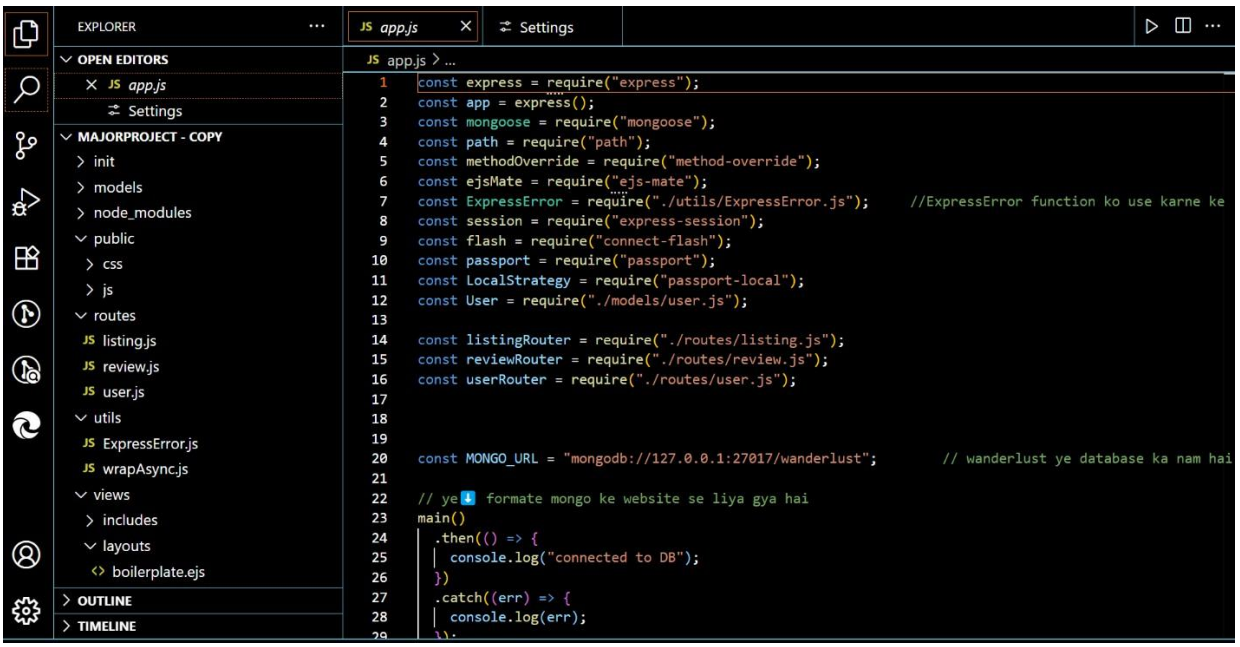
Website Description – ApnaRoom.com

ApnaRoom.com is a user-friendly online platform designed to help people find and book rooms for rent on a **daily, weekly, or monthly** basis. Whether you are a student, a working professional, or a traveler looking for short-term or long-term accommodation, ApnaRoom.com makes the process simple and hassle-free.

Users can easily browse available rooms in their desired location, view details like room size, amenities, rent, and photos, and contact the owner directly. If they are interested, they can also schedule a **visit to the room** before booking.

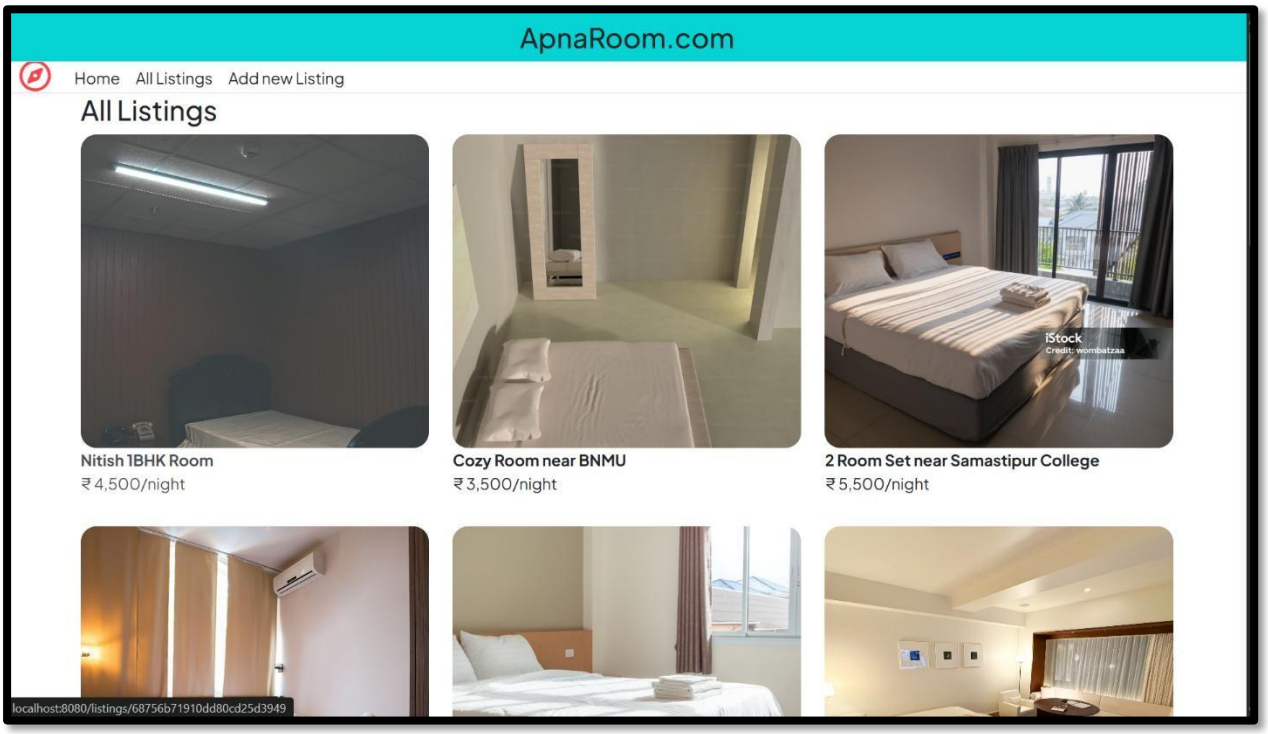
ApnaRoom.com is the perfect solution for anyone looking for **affordable and flexible room rentals** with the option to **visit before booking**.

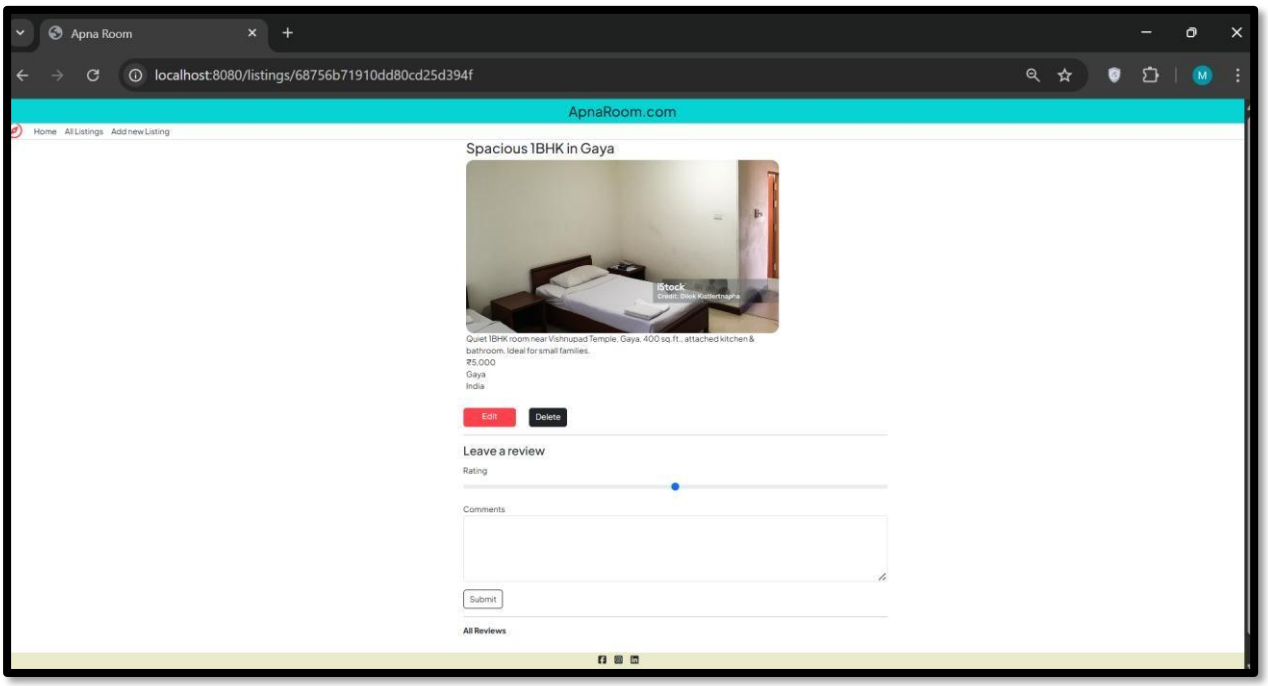
Project Structure:-



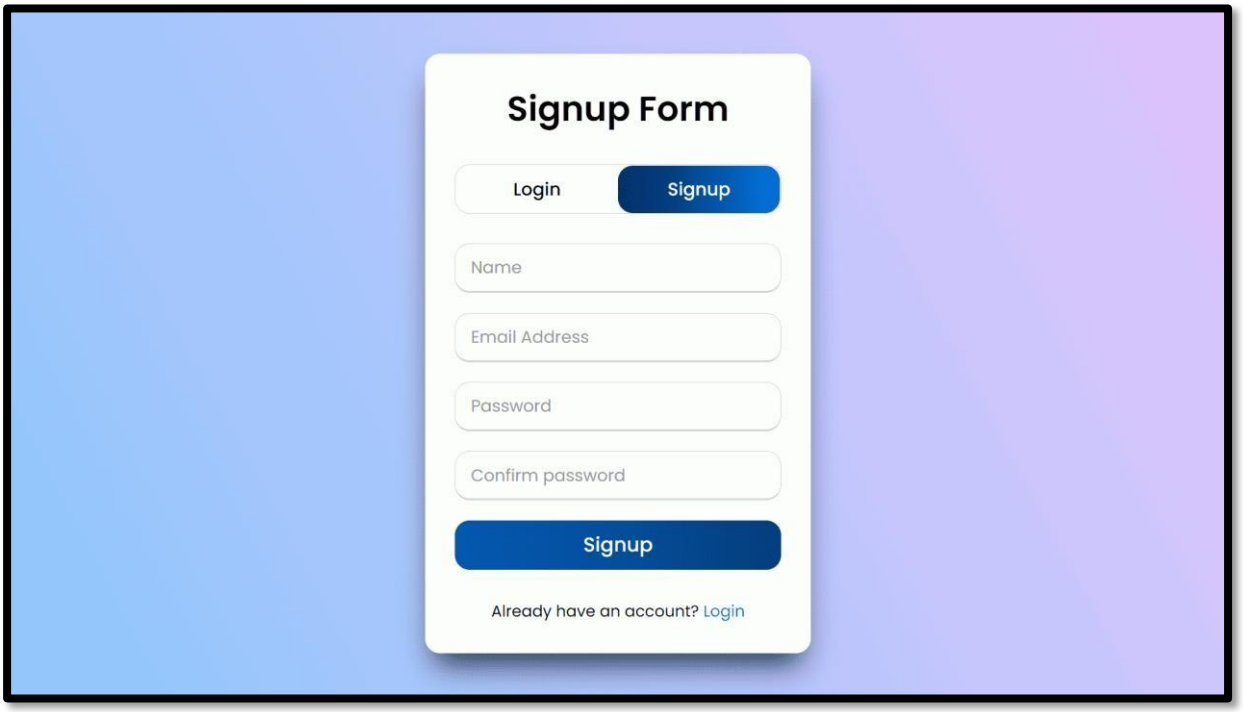
Frontend Part :-

User Interface(UI):-

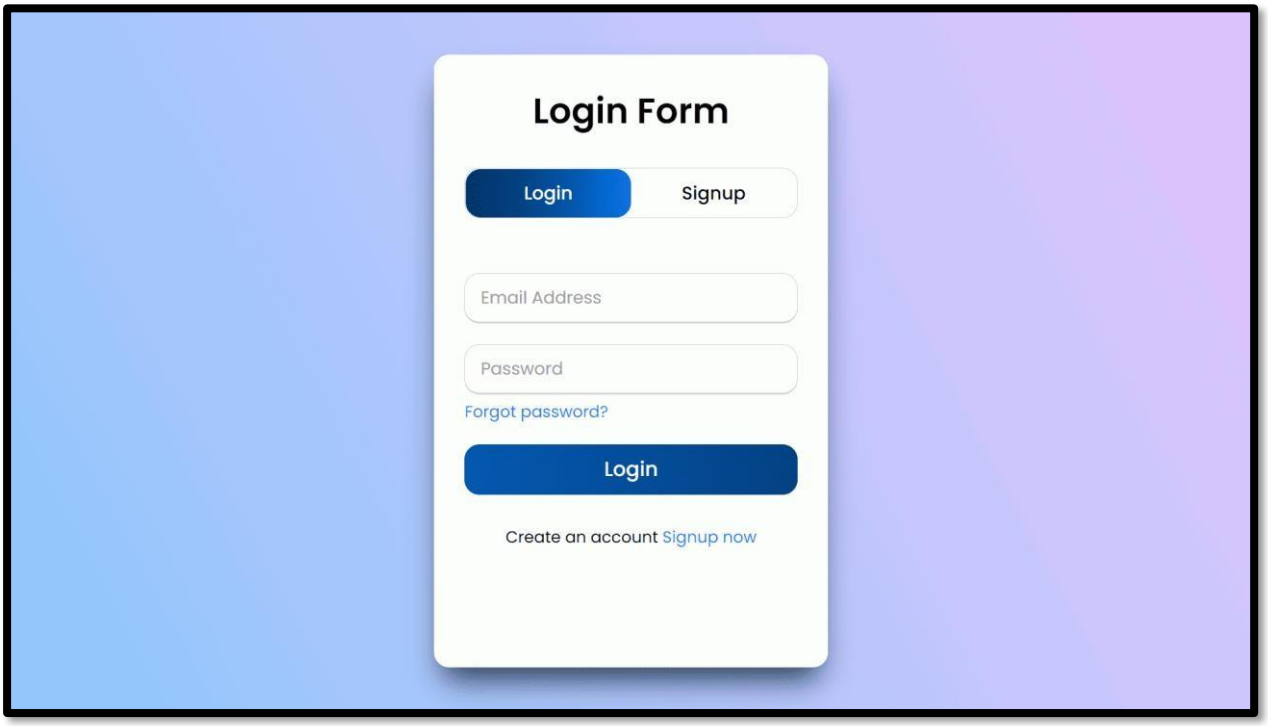


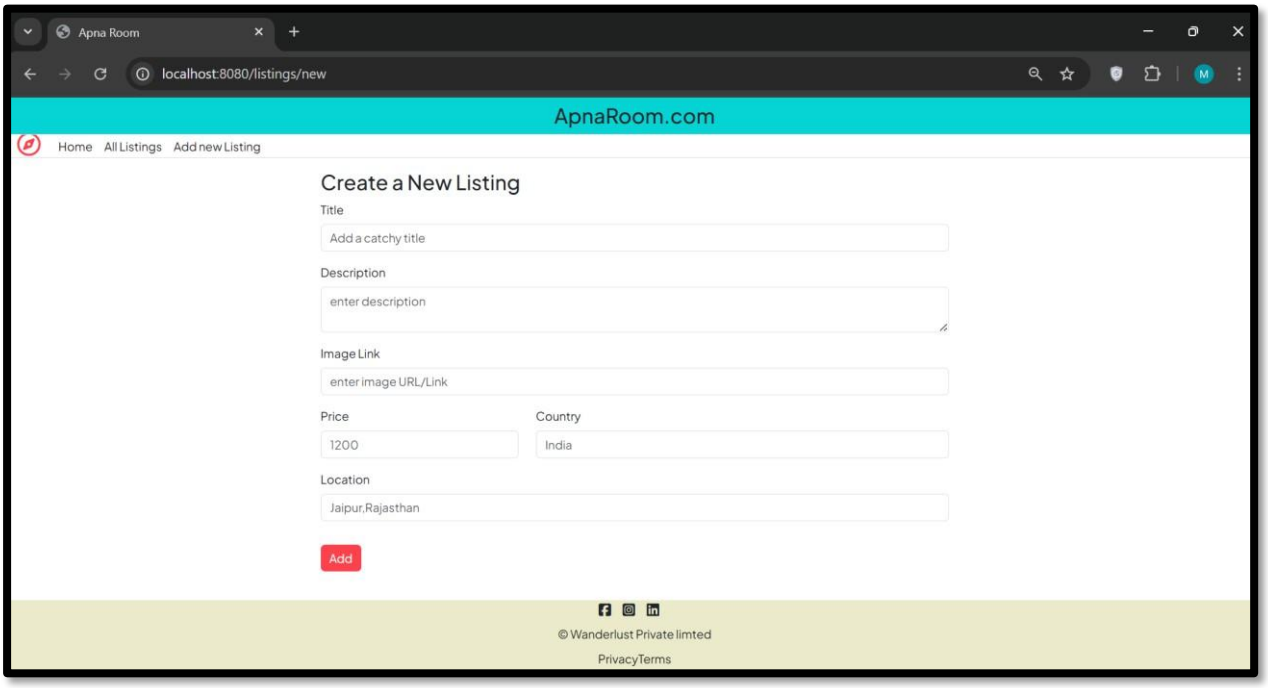


Signup form:-



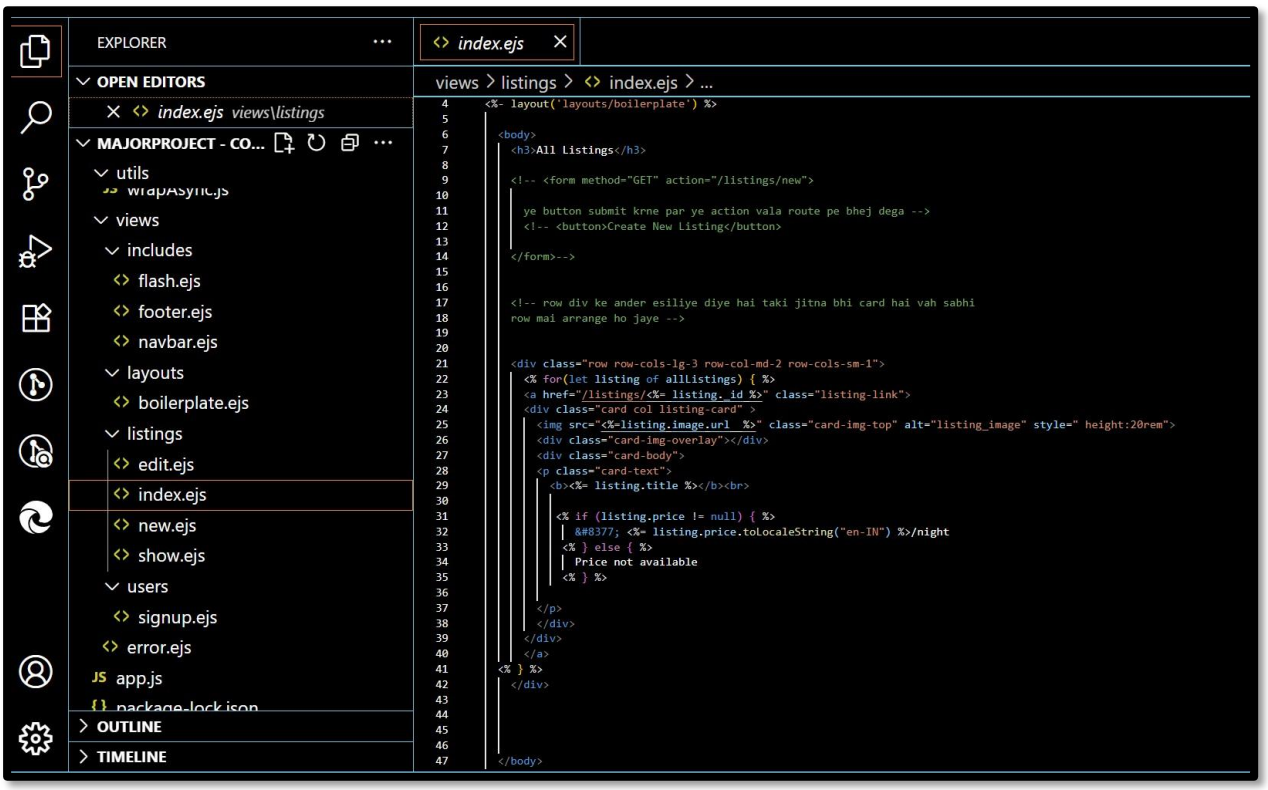
Signup form:-



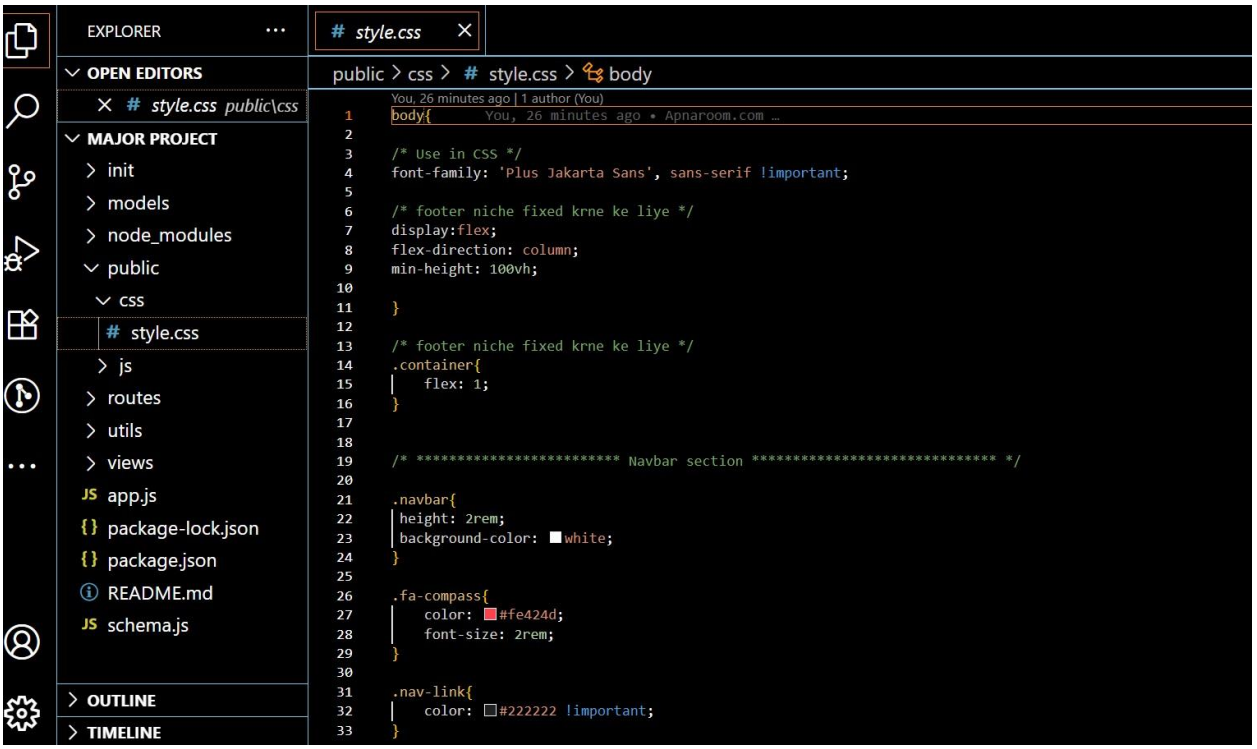


Some Important code:-

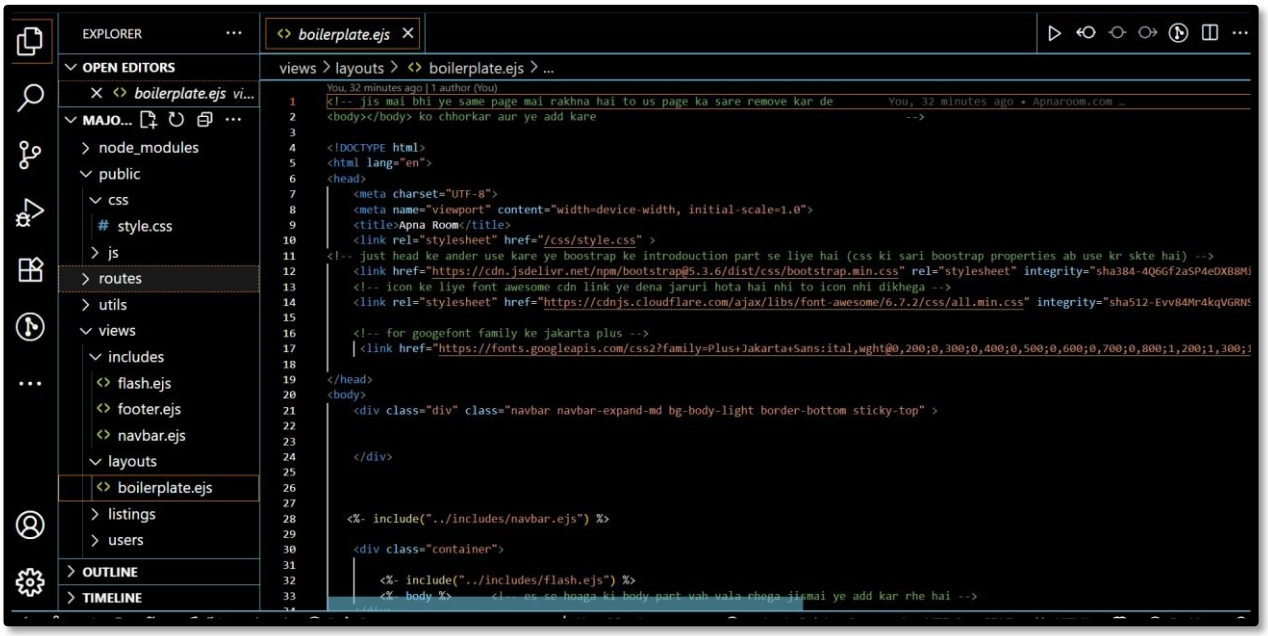
Index.ejs:-



styling:-



styling:-



Backend Part :-

Node.js & Express :-

```
listing.js routes
MAJO...
node_modules
public
  css
    style.css
  js
routes
  listing.js
  review.js
  user.js
utils
views
  includes
    flash.ejs
    footer.ejs
    navbar.ejs
  layouts
OUTLINE
TIMELINE
```

```
26
27
28  //index routes
29  router.get("/", wrapAsync(async (req, res) => {
30      const allListings = await Listing.find({});
31      res.render("listings/index.ejs", { allListings });
32  }));
33
34  //New Route
35  router.get("/new", (req, res) => {
36      res.render("listings/new.ejs");
37  });
38
39  //Create Route
40  router.post("/",validateListing, wrapAsync(async (req, res,next) => {
41          const newListing = new Listing(req.body.listing);
42          await newListing.save();
43          req.flash("success", "New Listing Created!");
44
45          res.redirect("/listings");
46
47   }));
48
49
50   //show routes
51   router.get("/:id", wrapAsync(async (req, res) => {
52    let { id } = req.params;
53    // yeha populate use krne se review ka id hi nhi dikhega sab chij dekhega
```

MongoDB & React :-

```
18
19
20  const MONGO_URL = "mongodb://127.0.0.1:27017/wanderlust";
21
22  // connect to mongoDB
23  main()
24    .then(() => {
25      console.log("connected to DB");
26    })
27    .catch((err) => {        You, 41 minutes ago • Apnaroom.com
28      console.log(err);
29    });
30
31  async function main() {
32    await mongoose.connect(MONGO_URL);
33  }
34
```

CONCLUSION

As a Web Developer Intern at **Invigo Info Tech**, I worked on enhancing my technical skills along with personal and industrial growth through real-time experience in a company environment. During this internship, I worked on multiple projects that helped me improve my knowledge in **HTML, CSS, JavaScript, Bootstrap, DBMS, MongoDB, and React**.

Using these technologies I developed my own project – **ApnaRoom.com**, a website where users can **search and find rooms on rent** for **1 day, 1 week, 1 month, or longer durations**. The website has a clean and responsive user interface that allows users to view room details and even visit the location before booking. Through this platform, I also showcased my skills, projects, education, certificates, and more.

Developing this website helped me strengthen my **logic building**, **problem-solving abilities**, and learn the importance of clean UI/UX. Creating features like the **landing page**, dynamic listings, and filters gave me deep practical knowledge of web development.

Overall, this internship at Invigo Info Tech and the development of ApnaRoom.com have played a key role in shaping my career and preparing me for the professional world of web development.

REFERENCES

References:-

- **INVIGO INFOTECH**
- Apna college
- Lovebabbar YouTube Channel

.

THANK YOU