LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(An Autonomous Institution since 2010)

Accredited by NAAC with 'A' Grade & NBA (Under Tier - I),

An ISO 21001:2018, 14001:2015, 50001:2018 Certified Institution

Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada

L.B. REDDY NAGAR, MYLAVARAM, NTR DIST., A.P.-521 230.

hodcse@lbrce.ac.in, cseoffice@lbrce.ac.in, Phone: 08659-222 933, Fax: 08659-222931

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Report on one-week Industrial Project Training on Application Development in "FULL STACK"

Event Type	: Industrial Project Training on Application Development in Full Stack
Dates	: 12/12/2024 to 18/12/2024
Time	: 09:00AM TO 04:00pm
Organized by	: Department of Computer Science and Engineering, LBRCE.
Name of the Industry	: TARAS System Solutions
Coordinator	: Mr. P. Veera Swamy
Total no. of Students attended	: 76

Objective

The objectives of a **Full Stack Development** Workshop are typically designed to provide participants with a comprehensive understanding of both front-end and back-end development, enabling them to build complete web applications. The workshop aims to help students understand the concepts of Full Stack Development, their significance, and their growing impact on various industries. By engaging in practical activities such as working with HTML, CSS, JavaScript, Node.js, databases like MySQL, MongoDB, modern front-end frameworks like React, Vue.js, or Angular. and connecting front-end and back-end through API calls to fetch and manipulate data, students gain experience in solving real-world problems using full stack technologies. The workshop encourages critical thinking and problem-solving by guiding students to identify and address real-life challenges using full stack.

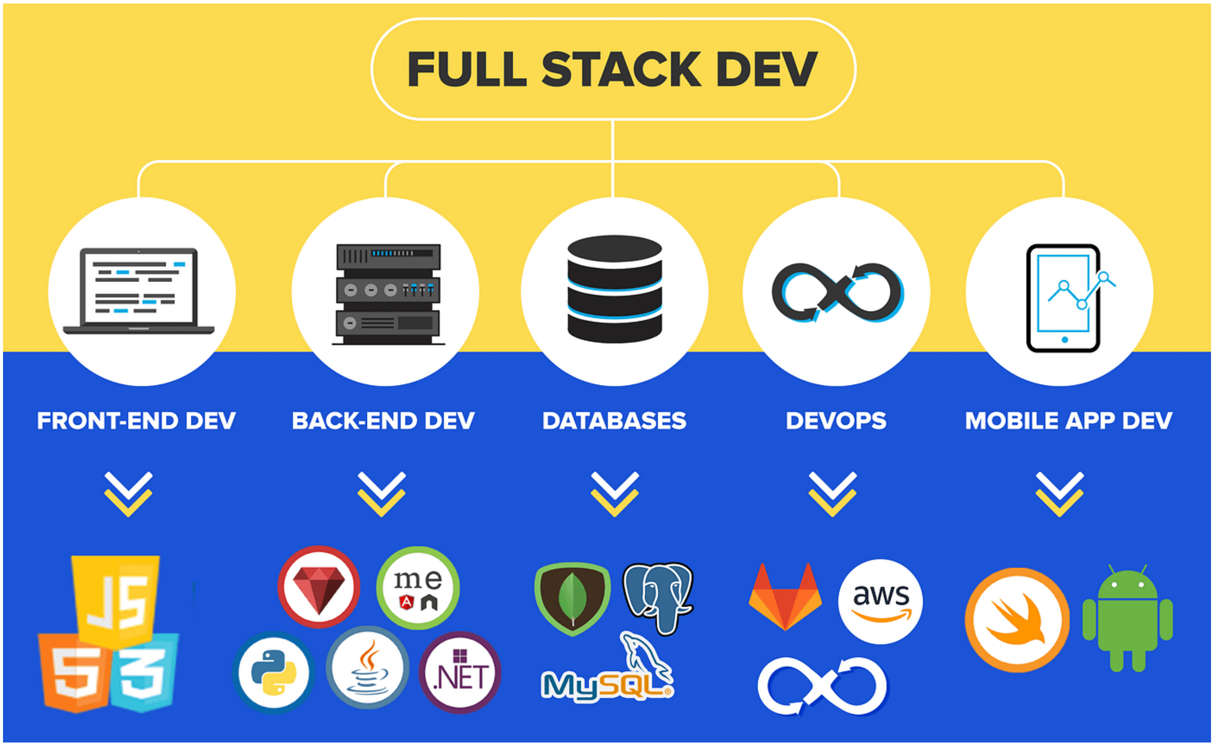
Impact Analysis

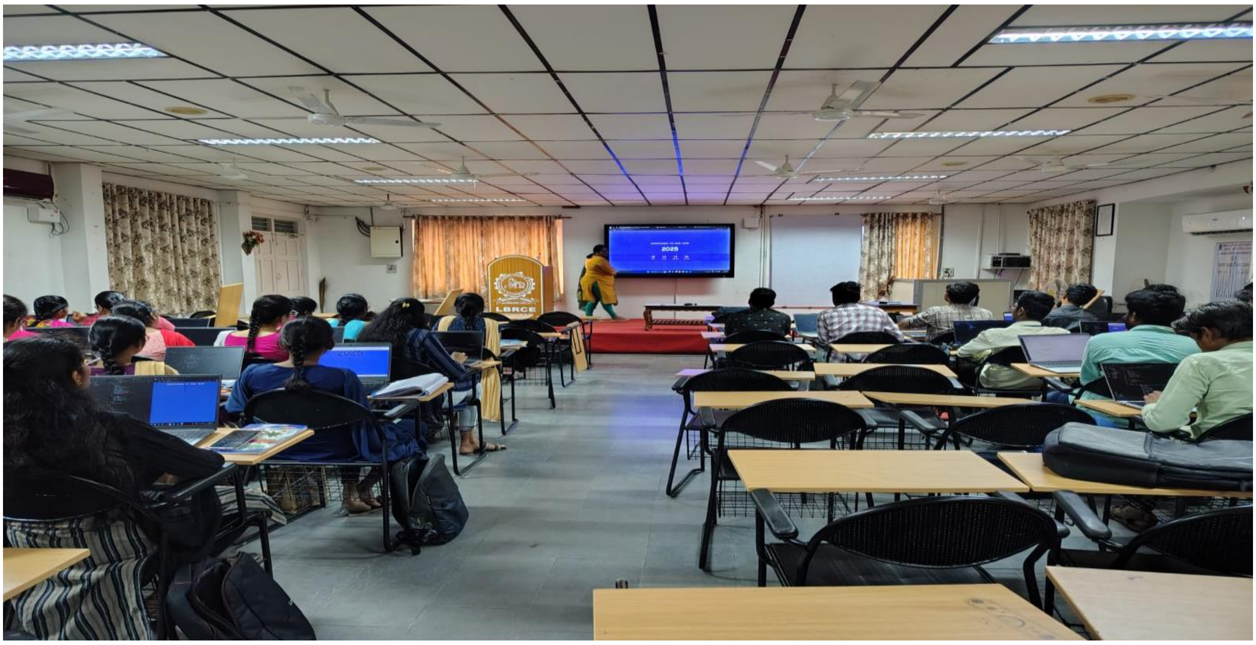
76 final-year **CSE** students from the 2021 batch attended a workshop on application development using **Full Stack**. Among them, **46 students** secured **16-week internship** for their current semester. These initiatives have greatly contributed to the student's academic progress and career preparedness while reinforcing the institution's dedication to industry-relevant education.

Day 1: Introduction to Full Stack, Applications of Full Stack

Full Stack:

- Introduction to Full Stack.
- Introduction to Front-End and Back-End.
- Introduction to Databases and Devops.
- Applications for Full Stack.





Day 2: Introduction to Front-end Development and Applications

The front-end of a full-stack web application refers to the part of the application that users interact with directly. It involves everything that users experience visually and interact with on a webpage, including text, images, buttons, menus, and other elements that make the application functional and visually appealing.

Key Components of Front-End Development:

1. **HTML (HyperText Markup Language):**
   - The backbone of web development.
   - Defines the structure of web pages using a system of tags and attributes.
   - Elements like headings, paragraphs, images, and links are all represented in HTML.
2. **CSS (Cascading Style Sheets):**
   - Used to style the HTML structure.
   - Defines how the elements should look (e.g., colors, fonts, layouts, spacing, and responsiveness).
   - CSS makes a webpage visually attractive and adapts it for different screen sizes (responsive design).
3. **JavaScript:**
   - The programming language that adds interactivity to a webpage.
   - Allows users to interact with the page dynamically (e.g., click events, form validation, animations).
   - Modern JavaScript frameworks like React, Vue.js, and Angular make front-end development more efficient by offering pre-built components and tools.
4. **Frameworks and Libraries:**

   - React.js: A JavaScript library for building user interfaces. It uses components to create reusable and dynamic UI elements.
   - Vue.js: A progressive JavaScript framework used for building interactive UIs and single-page applications.
   - Angular: A robust front-end framework that helps developers build dynamic web apps by combining HTML, CSS, and JavaScript.

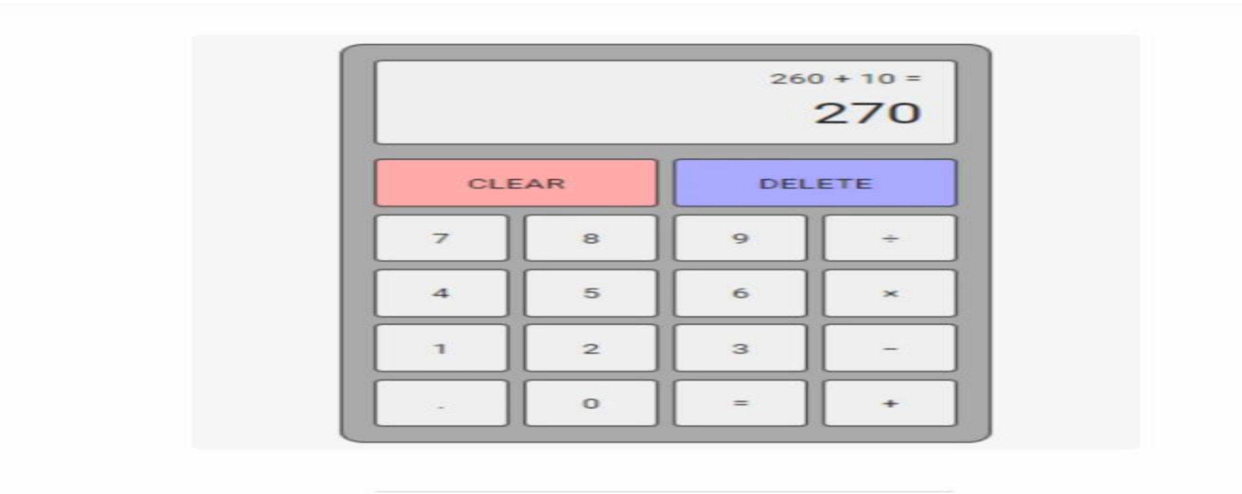
5. **Responsive Design:**
   - Ensures that web pages function well on a variety of devices (desktops, tablets, smartphones).
   - Techniques like media queries in CSS allow developers to adjust layouts for different screen sizes.

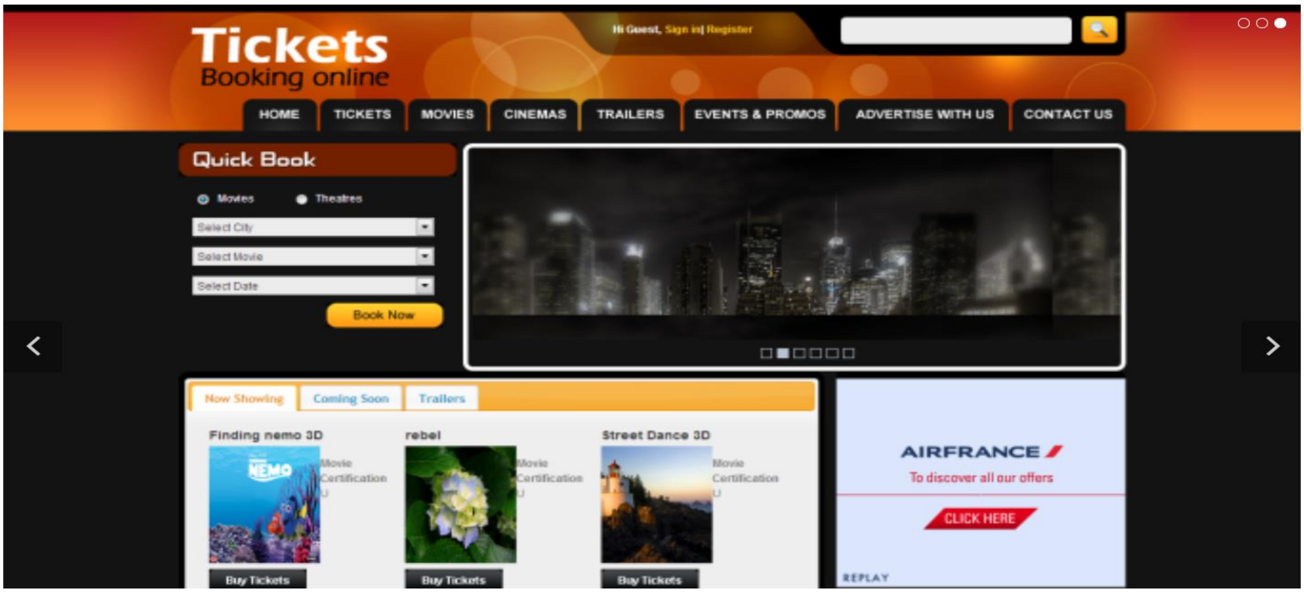
6. **Version Control:**
   - Git is a version control system that tracks changes in code and allows multiple developers to collaborate on the same project without conflicting changes.

Tools and Technologies in Front-End Development:

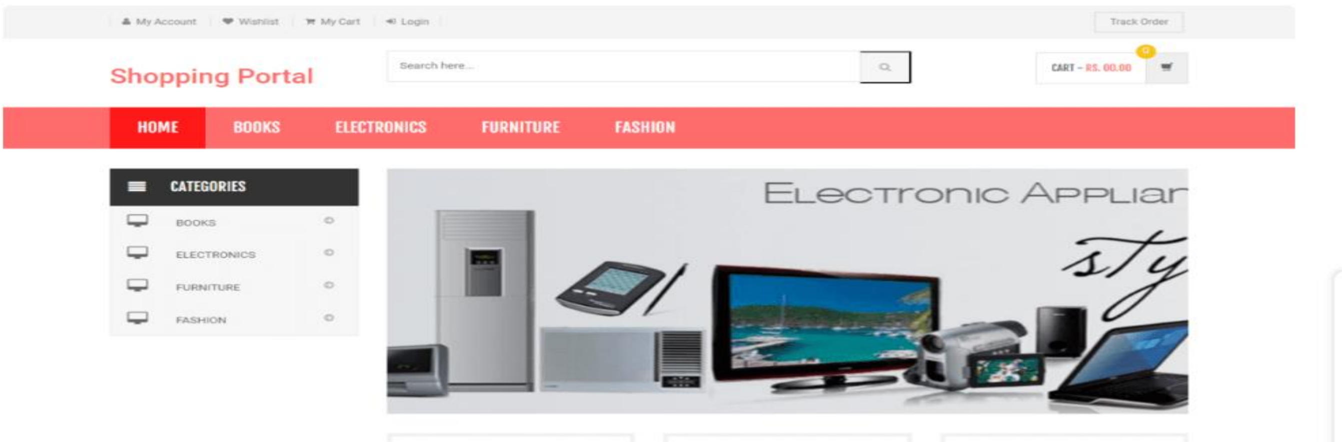
- **Text Editors:** Tools like Visual Studio Code, Sublime Text, and Atom are commonly used to write and manage code.
- **Browser Developer Tools**: Browsers like Chrome and Firefox provide built-in tools to debug and inspect code live on a webpage.
- **Package Managers:** Tools like npm (Node Package Manager) are used to manage libraries and dependencies.
- **Build Tools:** Tools like Webpack, Parcel, or Vite are used to bundle code and assets, improving performance.

Students Developed a websites:





Home Page



Day 3: Introduction to Mern Stack Technologies

The **MERN stack** is a popular set of technologies used for building modern web applications. It stands for **MongoDB**, **Express.js**, **React.js**, and **Node.js**. Each technology plays a key role in the full-stack development process, enabling developers to work in JavaScript for both the client-side (front-end) and server-side (back-end) of an application.

Here's a breakdown of each technology in the MERN stack:

1. MongoDB

- **Type**: NoSQL database
- **Role**: MongoDB is a document-oriented NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It allows for easy scaling and is designed for handling large amounts of unstructured or semi-structured data.
- **Key Features**:
  - Schema-less data storage, which means you can store records without needing to define a rigid structure beforehand.
  - Horizontal scaling through sharding, making it suitable for distributed applications.
  - Good for applications requiring high availability and speed.

2. Express.js

- **Type**: Web framework for Node.js
- **Role**: Express.js is a minimal, fast, and flexible web framework for building back-end applications and APIs with Node.js. It provides a set of tools to manage routing, middleware, request handling, and much more, making server-side development easier.
- **Key Features**:
  - Simplifies creating RESTful APIs and web applications.
  - Easy integration with databases, especially MongoDB.
  - Middleware support for handling requests, responses, authentication, etc.
  - Lightweight and unopinionated, allowing developers to customize the framework.

3. React.js

- **Type**: JavaScript library for building user interfaces (front-end)
- **Role**: React.js is a JavaScript library developed by Facebook for building dynamic and interactive user interfaces, typically for single-page applications (SPAs). React uses a component-based architecture, making it easy to create reusable UI components and manage the state of the application.

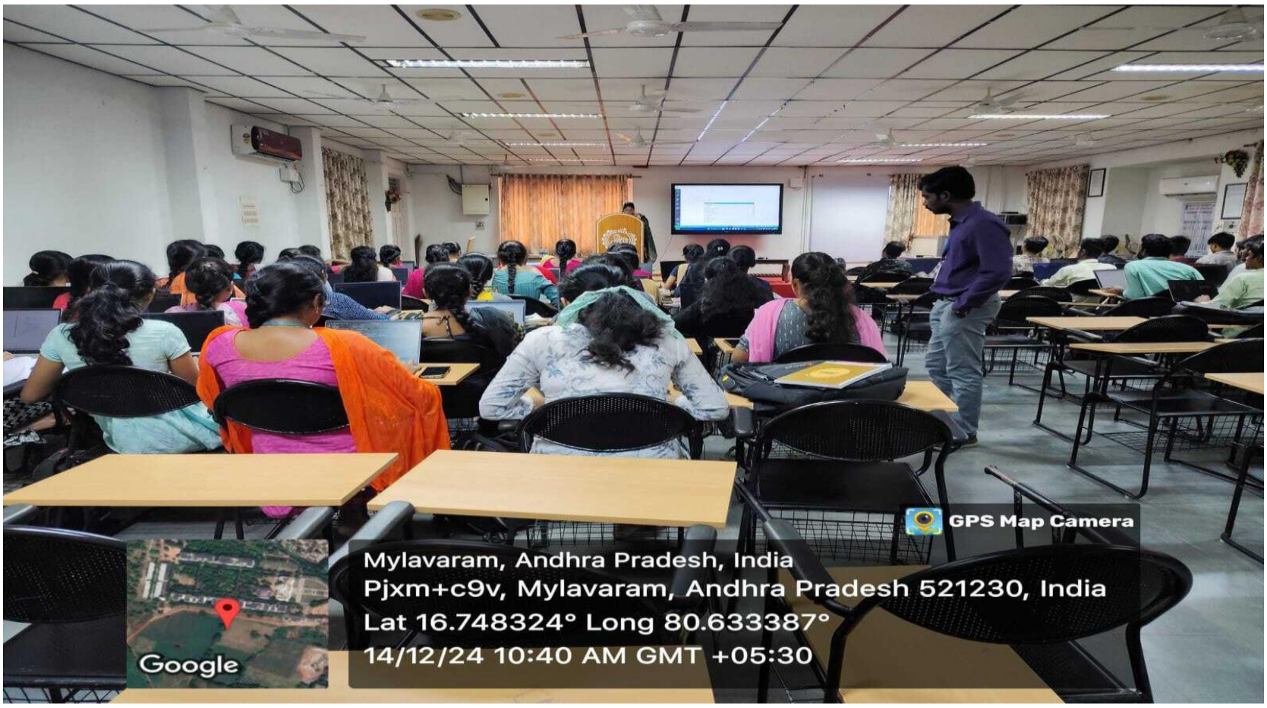
- **Key Features**:
    - Virtual DOM (Document Object Model), which enhances performance by minimizing direct manipulation of the real DOM.
    - Declarative syntax that makes it easy to define how the UI should look for any given state.
    - Ecosystem of tools like **React Router** for navigation and **Redux** for state management.
    - Works well with other front-end technologies and APIs, including back-end services.

4. Node.js

- **Type**: JavaScript runtime environment
- **Role**: Node.js allows developers to run JavaScript on the server side. It's built on the V8 JavaScript engine and enables asynchronous, event-driven architecture, which makes it highly efficient for building scalable network applications like web servers.
- **Key Features**:
    - Non-blocking, asynchronous I/O model, making it ideal for I/O-heavy applications such as APIs and real-time applications.
    - Large ecosystem of packages available through **npm** (Node Package Manager), making development faster and easier.
    - Single language (JavaScript) used for both client-side and server-side, improving consistency in development.

Benefits of Using the MERN Stack:

- **Full JavaScript Stack**: Since both the front-end and back-end use JavaScript, developers can work with one language across the entire stack, making development and code sharing easier and more efficient.
- **Fast Development**: The MERN stack has a rich ecosystem and a wide range of pre-built modules (especially via npm) that speed up the development process.
- **Scalability**: MongoDB, Node.js, and Express.js all support building scalable, high-performance applications, making the MERN stack ideal for web apps that may need to grow rapidly.
- **Active Community**: Since React.js, Node.js, and MongoDB are widely used, there is a large community with plenty of tutorials, documentation, and open-source libraries.
- **Real-Time Data Handling**: Node.js is well-suited for real-time applications, and React.js provides smooth interactions, making it a great choice for applications like chat systems, online gaming, or live feeds.



Day 4: Introduction to Back-end Development and Applications

The back-end of a web application refers to the server-side part that processes requests, handles business logic, interacts with databases, and serves data to the front-end (client-side). While the front-end is responsible for the user interface (UI) and user experience (UX), the back-end ensures that everything functions correctly behind the scenes to deliver the data or perform the operations needed.

In a full-stack application, the back-end is the core that supports all the functionality, such as data management, user authentication, payment processing, and more. It communicates with the front-end through APIs (Application Programming Interfaces) to deliver information that is displayed to the user.

Key Components of Back-End Development:

1. **Server:**
    - The server is where your application runs. It processes incoming requests from users (typically via HTTP/HTTPS), performs required actions, and returns responses.
    - Servers can be physical machines, virtual machines, or cloud-based instances (e.g., AWS, Azure, Google Cloud).
    - Common server software includes Apache, Nginx, and IIS.
2. **Database:**
    - Databases store and organize data for the application. They are queried by the back-end to retrieve, store, update, or delete data.
    - Relational Databases (SQL): These use structured query language (SQL) to define and manipulate data (e.g., MySQL, PostgreSQL).
    - NoSQL Databases: These store data in more flexible, non-tabular formats, such as

documents, key-value pairs, or graphs (e.g., MongoDB, Cassandra).

3. **Server-Side Programming Languages:**
   - Back-end development relies on server-side programming languages to implement business logic, process requests, and interact with databases.
   - Some common back-end programming languages include:
     - JavaScript (with Node.js)
     - Python (with frameworks like Django, Flask)
     - Java (with Spring, Hibernate)
     - Ruby (with Ruby on Rails)
     - PHP (with Laravel)
     - C# (with .NET)
4. **Web Frameworks:**
   - A web framework is a tool that helps developers build web applications more easily by providing pre-built components for routing, handling requests, and managing data flow.
   - Popular web frameworks for back-end development include:
     - Express.js (for Node.js)
     - Django (for Python)
     - Flask (for Python)
     - Spring Boot (for Java)
     - Ruby on Rails (for Ruby)
     - Laravel (for PHP)
5. **APIs (Application Programming Interfaces):**
   - APIs allow the back-end to interact with the front-end and other services. Typically, back-end systems expose data and functionality via RESTful APIs or GraphQL APIs.
   - REST: A common architectural style that uses standard HTTP methods like GET, POST, PUT, and DELETE to interact with resources.
   - GraphQL: A more flexible alternative to REST that allows clients to request only the data they need, rather than receiving fixed responses.
6. **Authentication and Authorization:**
   - Authentication ensures that the user is who they claim to be (e.g., via login credentials like username and password).
   - Authorization determines what resources or actions a user is permitted to access or perform based on their identity (e.g., user roles like admin or guest).
   - Common authentication techniques include JWT (JSON Web Tokens), OAuth, and session-based authentication.

7. **Middleware:**
   - Middleware functions are intermediary processes that handle requests before they reach the actual endpoint or after the response is sent back to the user.
   - Middleware can handle tasks such as logging, validation, error handling, authentication, and more.
   - In Express.js, for example, middleware is a central part of how requests are processed.

Back-End Technologies:

1. **Databases**:
   - **MySQL**, **PostgreSQL**, **SQLite** (SQL databases).
   - **MongoDB**, **Cassandra**, **Redis** (NoSQL databases).
2. **Web Servers**:
   - **Apache**: A widely used open-source web server that serves HTTP content.
   - **Nginx**: A high-performance web server often used for serving static files and reverse proxying.
3. **Frameworks and Libraries**:
   - **Node.js/Express.js** (JavaScript-based back-end development).
   - **Django/Flask** (Python-based frameworks).
   - **Spring Boot** (Java-based framework).
   - **Ruby on Rails** (Ruby-based framework).
   - **Laravel** (PHP-based framework).
4. **Authentication Libraries**:
   - **Passport.js** (for Node.js).
   - **JWT (JSON Web Tokens)**: A popular method for token-based authentication.
   - **OAuth**: A protocol for authorization commonly used in social logins (e.g., "Login with Google" or "Login with Facebook").
5. **Task Scheduling and Queues**:
   - **RabbitMQ**, **Kafka**, **Bull.js**: Used for queuing tasks and managing background processes (e.g., email sending, data processing).

Day 5: Communication Between Front-end and Back-end Development

In modern web development, the front-end (client-side) and back-end (server-side) need to communicate with each other to create dynamic, interactive applications. The front-end handles the user interface and user experience, while the back-end is responsible for data processing, storage, and business logic.

The communication between these two layers is essential for web applications to function properly, and it typically happens through API calls or HTTP requests. Below, we'll explain how this communication works, the methods involved, and common technologies used.

How Communication Works:

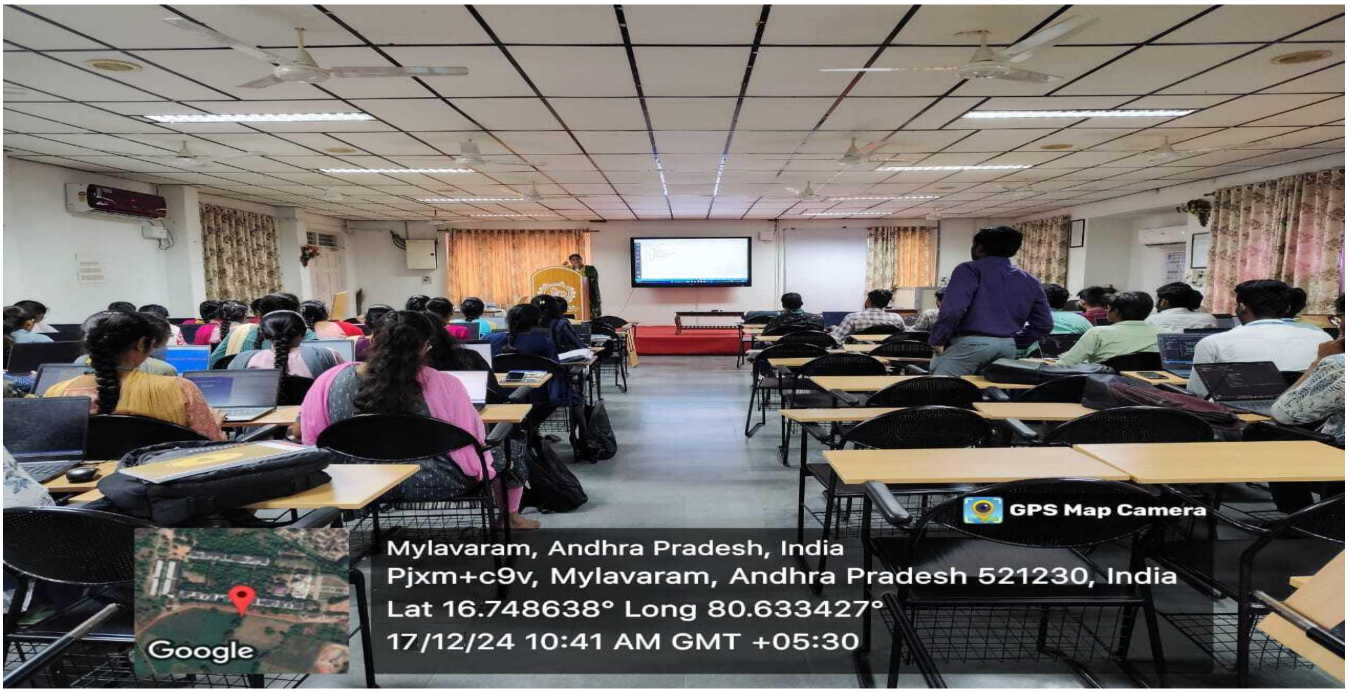
1. **User Interaction:**
   - The process begins when a user interacts with the front-end of the application (e.g., clicks a button, submits a form, or loads a page).
2. **Front-End Sends HTTP Requests:**
   - The front-end sends an HTTP request (usually using AJAX, fetch, or Axios) to the back-end. The request is sent to a specific API endpoint (a URL) on the back-end server.
   - The request can contain data sent from the front-end (e.g., form data, JSON objects, or query parameters).
   - This HTTP request typically follows one of the following HTTP methods:
     - GET: To request data (e.g., fetching user information).
     - POST: To send new data (e.g., submitting a form or creating a new resource).
     - PUT: To update existing data (e.g., updating a user's profile).
     - DELETE: To delete data (e.g., removing an item from a shopping cart).
3. **Back-End Processes the Request:**
   - The back-end server (e.g., built with Node.js, Django, or Spring Boot) receives the request, processes the data, interacts with the database if necessary, performs business logic, and prepares a response.
   - The back-end might fetch, update, or delete records from a database (e.g., MongoDB, MySQL, or PostgreSQL) based on the request.
4. **Back-End Sends Response:**
   - After processing the request, the back-end sends an HTTP response back to the front-end.
   - The response typically contains:
     - Data (e.g., user details, list of products) in formats like JSON or XML.
     - Status Code to indicate the result of the request (e.g., 200 OK, 404 Not Found, 500 Internal Server Error).

- Headers containing metadata (e.g., content type, authorization tokens).

5. **Front-End Updates the UI:**
   - The front-end receives the response from the back-end and uses the data to update the user interface dynamically.
   - This could mean rendering new data on the page (e.g., displaying a list of products) or showing a success/error message based on the response status.

Common Communication Methods and Technologies:

1. **AJAX (Asynchronous JavaScript and XML):**
   - AJAX allows the front-end to send requests to the back-end and receive responses without refreshing the entire page.
   - AJAX is typically used with JavaScript and works asynchronously, meaning the page doesn't need to reload to process requests or display new data.
2. **Fetch API**:
- The **Fetch API** is a modern alternative to AJAX for making HTTP requests. It is promise-based, making it easier to work with asynchronous code.
3. **Axios**:
- **Axios** is a popular JavaScript library for making HTTP requests. It provides a simpler, more feature-rich API compared to Fetch and is often used in front-end applications.
4. **RESTful APIs (Representational State Transfer)**:
- A **RESTful API** is a set of guidelines for building APIs that work over HTTP, where each endpoint represents a specific resource, and the HTTP methods (GET, POST, PUT, DELETE) are used to perform operations on these resources.



Day 6: Project Assignment and Assessment

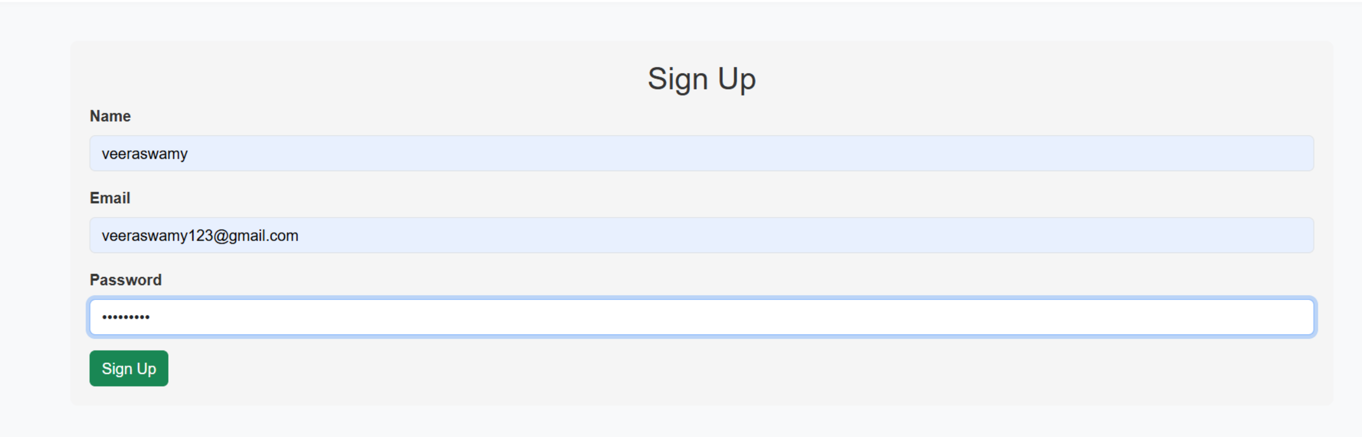
Project Assignment

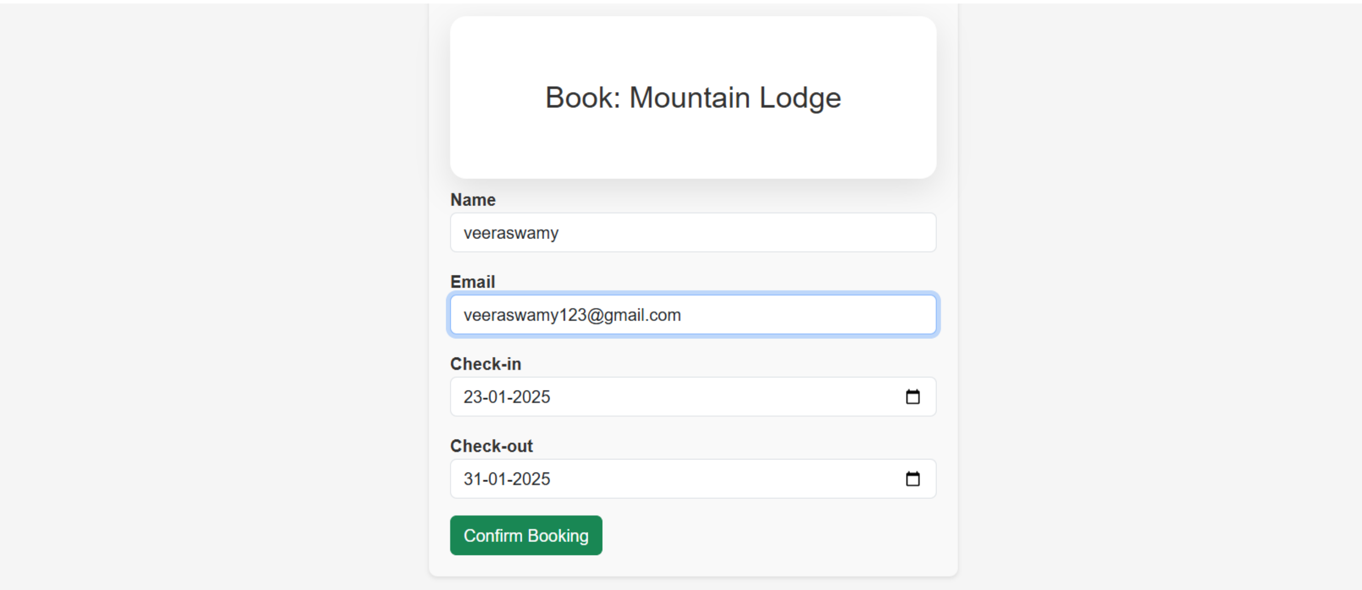
- Provided a list of 15 projects for participants to select and work on.
- Participants were required to demonstrate their progress to the instructor.

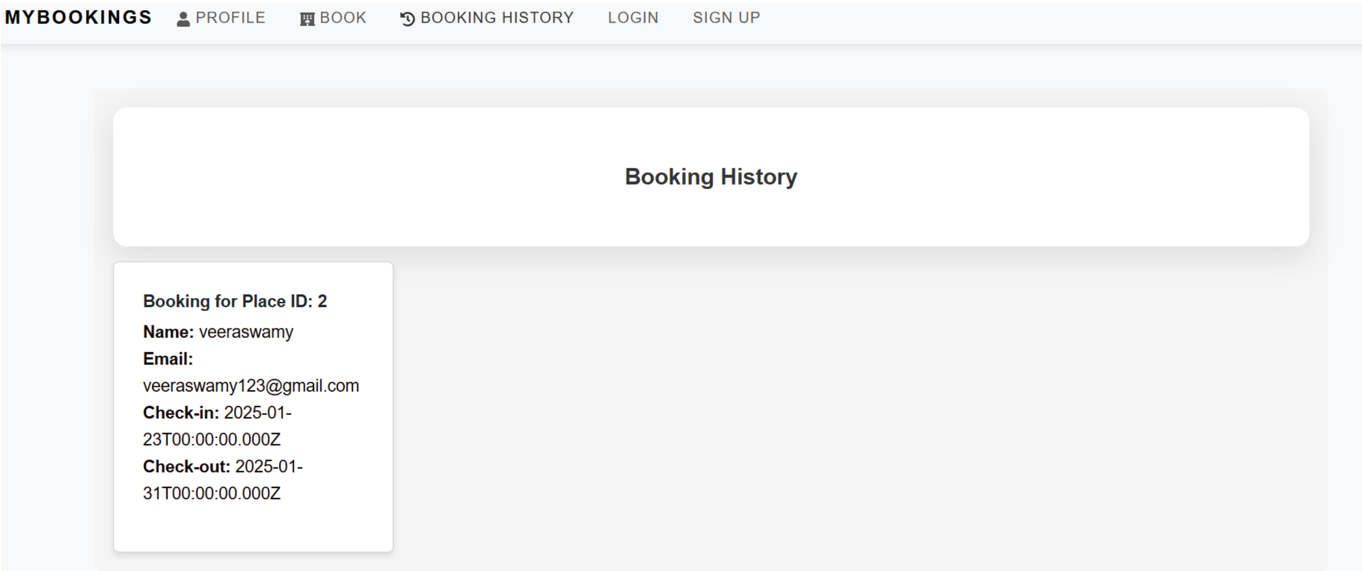
Afternoon Assessment

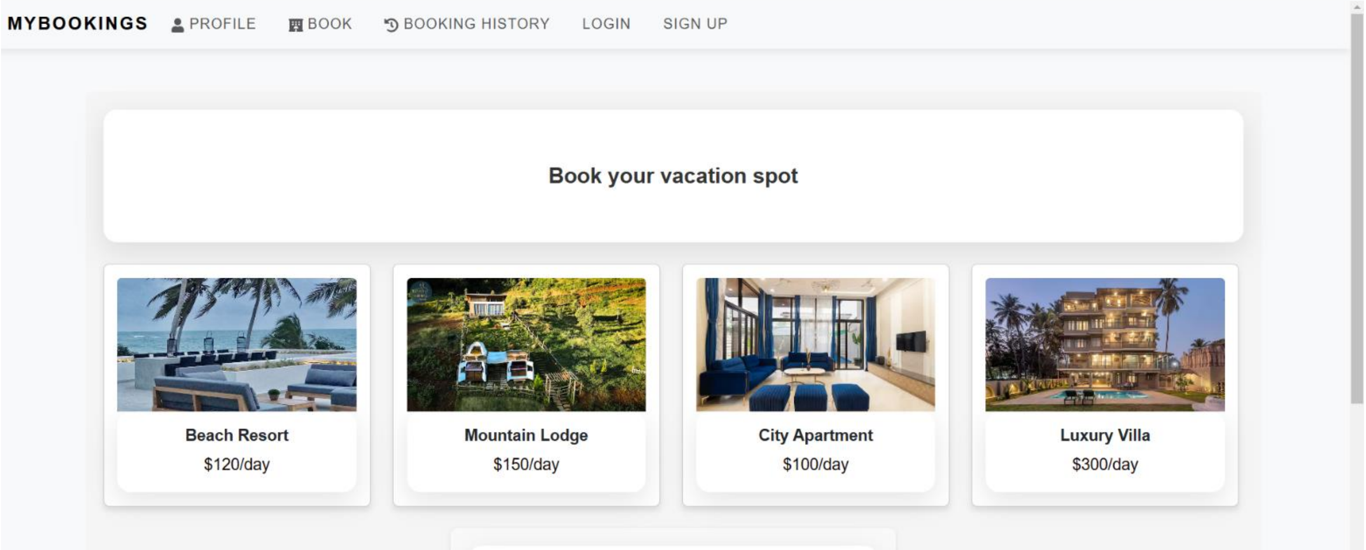
- Conducted an assessment to evaluate understanding and practical skills.
- Focused on implementation and problem-solving abilities.

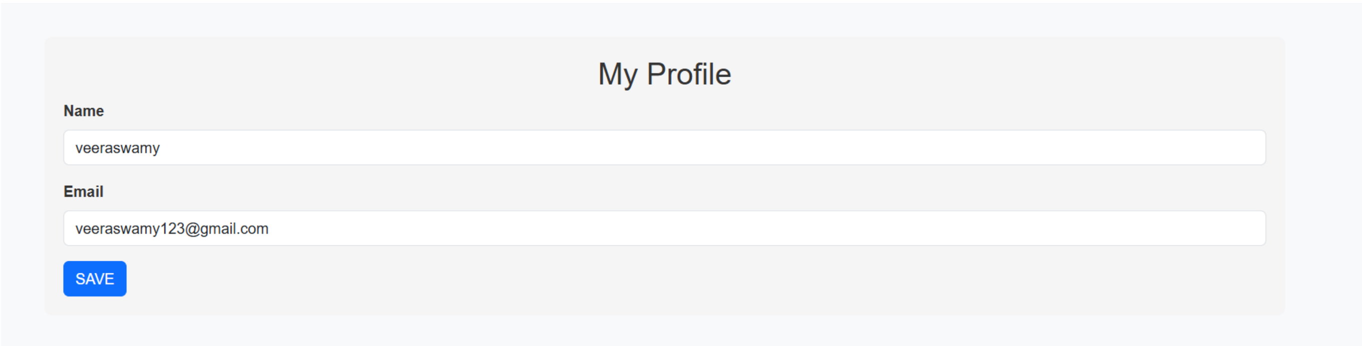
Sample Students Projects:

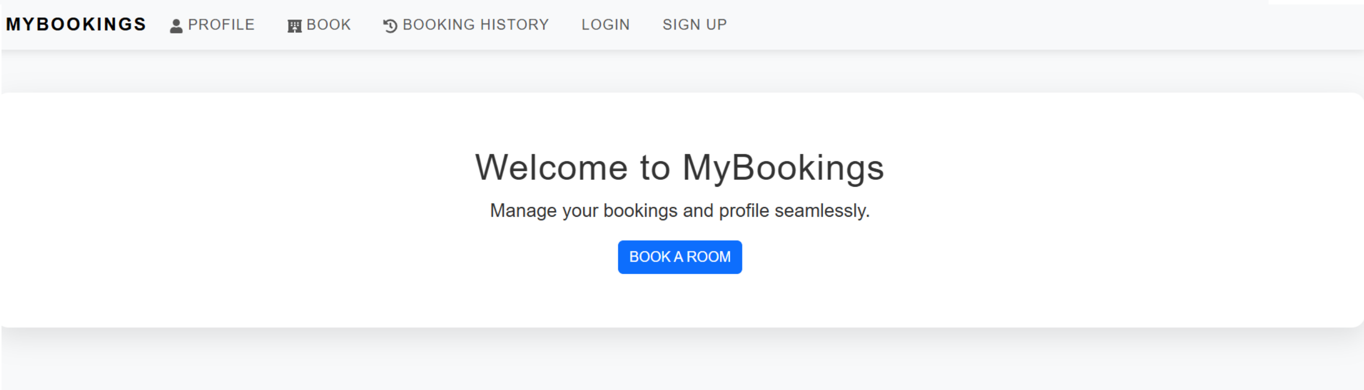


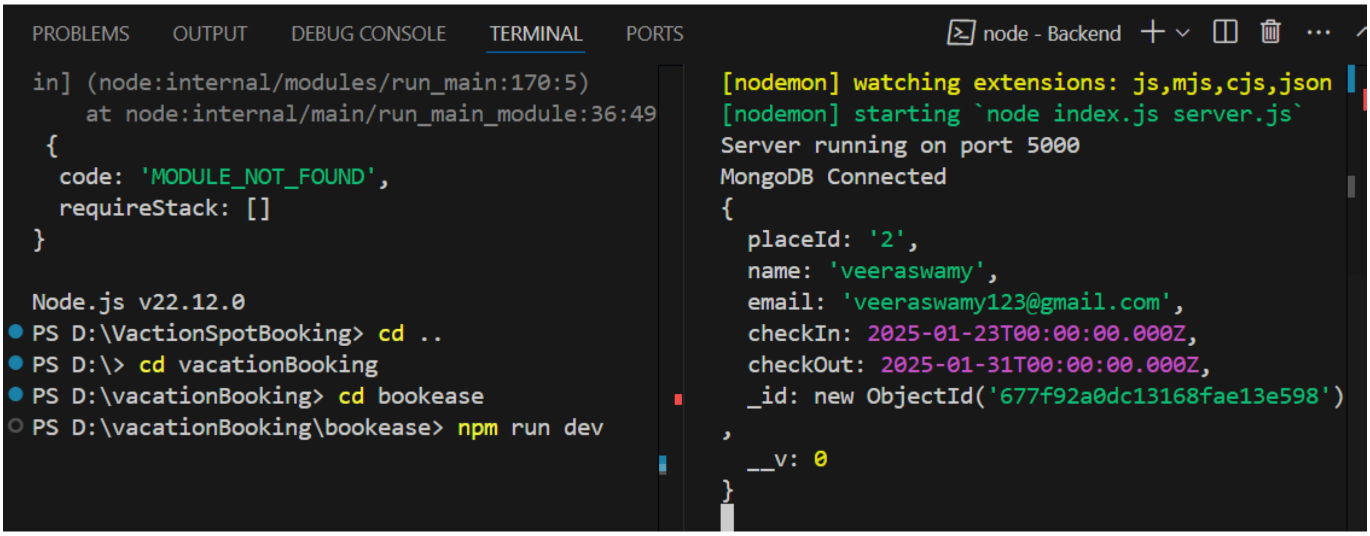


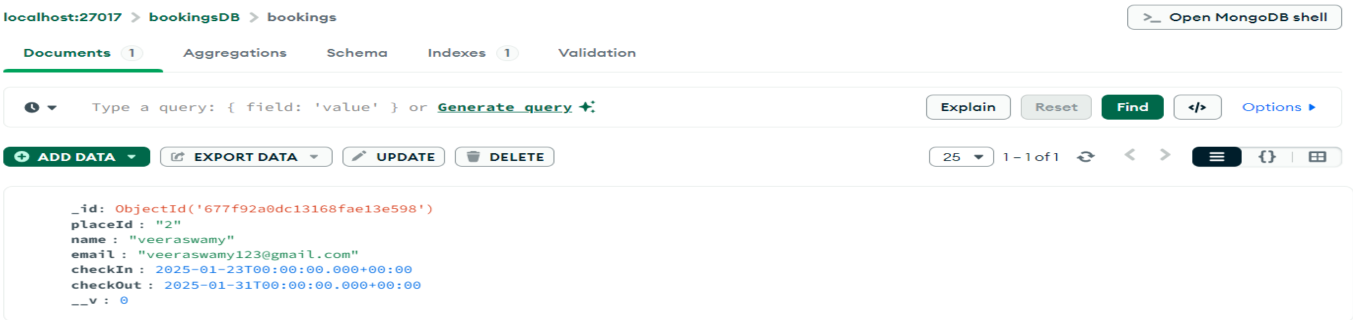

MYBOOKINGS
PROFILE
BOOK
BOOKING HISTORY
LOGIN
SIGN UP
Booking History
Booking for Place ID: 2
Name: veeraswamy
Email: veeraswamy123@gmail.com
Check-in: 2025-01-23T00:00:00.000Z
Check-out: 2025-01-31T00:00:00.000Z


MYBOOKINGS
PROFILE
BOOK
BOOKING HISTORY
LOGIN
SIGN UP
Book your vacation spot
Beach Resort
$120/day
Mountain Lodge
$150/day
City Apartment
$100/day
Luxury Villa
$300/day


My Profile
Name
veeraswamy
Email
veeraswamy123@gmail.com
SAVE


MYBOOKINGS
PROFILE
BOOK
BOOKING HISTORY
LOGIN
SIGN UP
Welcome to MyBookings
Manage your bookings and profile seamlessly.
BOOK A ROOM





OUTCOME:

The **outcome of a full-stack development workshop** includes:

- A **complete web application** with both front-end and back-end components.
- Hands-on experience with **modern development tools** and **best practices**.
- Knowledge of how to **deploy and host** a full-stack application.
- An understanding of how to handle **security**, **authentication**, and **data management.**
- Improved skills in **collaboration** and **version control** using **Git**.
- A tangible **project for your portfolio**, showcasing your ability to build full-stack web applications.
- Enhanced **confidence** in your ability to tackle future web development challenges.

Conclusion:

This workshop not only provides a deep dive into **full-stack web development** but also prepares participants for the real-world challenges of building and maintaining scalable, secure, and user-friendly web applications. By gaining both **front-end and back-end skills**, participants are equipped to approach development projects holistically, ensuring they are capable of building end-to-end solutions.

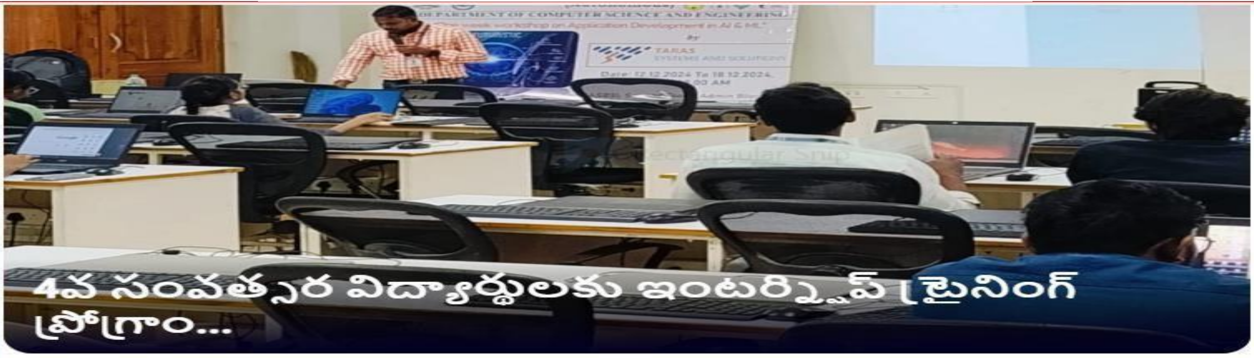
News Paper cuts:

సాక్షి

ఇంటర్న్‌షిప్‌పై శిక్షణ

మైలవరం: మైలవరం లకిరెడ్డి బాలిరెడ్డి ఇంజినీరింగ్ కళాశాలలో కంప్యూటర్ సైన్స్ విభాగం ఆధ్వర్యంలో ఇంటర్న్‌షిప్‌పై శిక్షణ కార్యక్రమం గురువారం ప్రారంభమైంది. కంప్యూటర్ సైన్స్ విభాగాధిపతి డాక్టర్ డి. వీరయ్య మాట్లాడుతూ ఇంటర్న్‌షిప్ అనేది విద్యార్థులకు విద్య, కెరీర్ మధ్య ఉన్న గ్యాప్‌ను తీర్చే ప్రధాన మార్గమన్నారు. టారాస్ సాఫ్ట్‌వేర్ సొల్యూషన్స్ సంస్థ ఆధ్వర్యంలో జరిగిన ఈ కార్యక్రమంలో తృతీయ సంవత్సరం సీఎస్ఈ విభాగాల్లో బ్లాక్ చైన్ టెక్నాలజీ, ఏఐ అండ్ ఎంఎల్, ఫుల్ స్టాక్ నాలుగో సంవత్సరం విద్యార్థులు పాల్గొన్నారు.

13/12/2024 | NTR(Tiruvuru) | Page : 9
Source : https://epaper.sakshi.com/



లకిరెడ్డి బాలిరెడ్డి కాలేజ్ ఆఫ్ ఇంజనీరింగ్ (LBRCE), మైలవరంలో కంప్యూటర్ సైన్స్ విభాగము లోడాక్టర్ డి.వీరయ్య పర్యవేక్షణలో టారాస్ సాఫ్ట్‌వేర్ సొల్యూషన్స్ వారు 3 సి.ఎస్.ఇ (CSE) విభాగాల్లో (బ్లాక్‌చైన్ టెక్నాలజీ, ఏఐ & ఎంఎల్ మరియు ఫుల్ స్టాక్) 4వ సంవత్సర విద్యార్థులకు ఇంటర్న్షిప్ ట్రైనింగ్ 12/12/24 నుంచి 18/12/24 వరకు ఈ ప్రోగ్రాం నిర్వహిస్తున్నారు. ఇంటర్న్షిప్ అనేది విద్యార్థులకు విద్య మరియు కెరీర్ మధ్య ఉన్న గ్యాప్‌ను తీర్చే ప్రధాన మార్గం. విద్యార్థులను

ప్లేస్‌మెంట్స్ కోసం మరింత అర్హులుగా నిలబెడతాయి అని కంప్యూటర్ సైన్స్ విభాగాధిపతి డాక్టర్ డి. వీరయ్య తెలిపారు.

#AndhraPradesh

Mr.P.Veera Swamy	**Dr. D. Veeraiah**
(Coordinator)	**(Head of the Department)**