INDUSTRIAL INTERNSHIP REPORT ON

FULLSTACK DEVELOPMENT WITH MONGODB

Submitted in accordance with the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the faculty Guide ship of

Mr. CH. SAMSONU MTech.

Associate professor

Submitted by:

VANGA NAVYA

Reg.No 218X1A05I6

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

ACCREDITED BY NAAC WITH 'A' GRADE

(APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA)

NH-5, CHOWDAVARAM, GUNTUR – 522019.

2021 - 2025

STUDENT'S DECLARATION

I**, VANGA NAVYA (218X1A05I6)** a student of **Bachelor of Technology** program of the department of **COMPUTER SCIENCE AND ENGINEERING** do hereby declare that I have completed the mandatory industrial internship from **15-07-2024 to 02-11-2024** in **SMART INTERNZ** under the faculty guidance of **Mr. CH. SAMSONU,** Associate Professor, Department of **COMPUTER SCIENCE AND ENGINEERING, KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY.**

(Signature and Date)

OFFICIAL CERTIFICATION

This is to certify that **VANGA NAVYA (218X1A05I6)** has completed his/her Industrial Internship in **SMART INTERNZ** on **FULLSTACK DEVELOPMENT WITH MONGODB** under my supervision as a part of partial fulfillment of the requirement for the Degree of Bachelor of Technology in the department of **COMPUTER SCINENCE AND ENGINEERING, KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY**.

This is accepted for evaluation.

(Signatory with Date and Seal)

Endorsements

Faculty Guide

Head of the Department

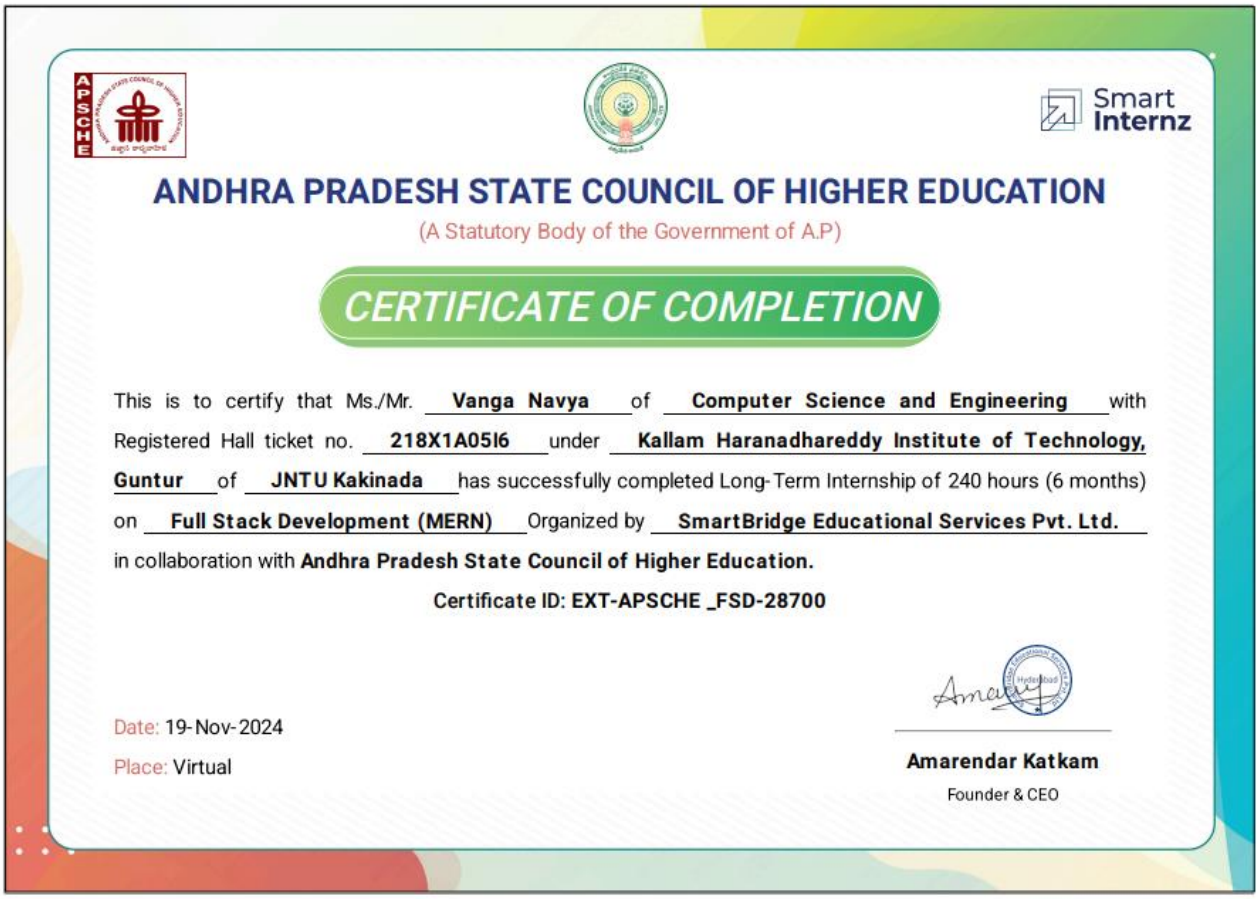
Principal

External Examiner

CERTIFICATE FROM INTERN ORGANIZATION

This is to certify that **VANGA NAVYA (218X1A05I6)** of Kallam Haranadhareddy Institute of Technology underwent industry internship in **SMART INTERNZ** from **15-07-2024 to 02-11-2024.** The overall performance of the intern during his/her internship isfound to be **Satisfactory** (Satisfactory / Not Satisfactory).

Authorized Signatory with Date and Seal



Smart Internz

ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Vanga Navya** of **Computer Science and Engineering** with Registered Hall ticket no. **218X1A05I6** under **Kallam Haranadhareddy Institute of Technology, Guntur** of **JNTU Kakinada** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

Certificate ID: EXT-APSCHE _FSD-28700

Date: 19-Nov-2024

Place: Virtual

Amarendar Katkam

Founder & CEO

ACKNOWLEDGMENT

We profoundly grateful to express our deep sense of gratitude and respect towards **Sri. KALLAM MOHAN REDDY, Chairman, KHIT, Guntur** for his precious support in the college.

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, Guntur for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY,** Principal, KHIT, Guntur for his support during and until the completion of the project.

We are greatly indebted to **Dr. V. RAJIV JETSON** Professor & Head of the department, **COMPUTER SCIENCE AND ENGINEERING,** KHIT, Guntur for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college during Industry Internship.

We are also thankful to our Project Coordinator **Mr. G. MANTHRU NAIK** who helped us in each step of our project.

We extend our deep sense of gratitude to our Internal Guide **Mr.CH. SAMSONU**, Associate Professor, and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of greathelp towards the successful completion of our project.

Place: Guntur VANGA NAVYA

Date 218X1A05I6

LIST OF CONTENTS

TITLE **PAGE NO**

CHAPTER 1: EXECUTIVE SUMMARY

In the SmartInternz full-stack development internship, participants typically work towards achievinga set of learning objectives that are designed to equip them with the skills and knowledge necessaryto succeed in the field of web development. Here are some common learning objectives and the corresponding outcomes achieved during the internship:

1. **Mastering Front-End Technologies:** Interns learn HTML, CSS, and JavaScript thoroughly, enabling them to create visually appealing and interactive user interfaces for webapplications.
2. **Understanding Back-End Development:** Through hands-on training in technologieslike Node.js, Express.js, and databases (e.g., MongoDB, SQL), interns gain proficiency in building robust server-side applications and managing data effectively.
3. **Full-Stack Integration**: Interns learn how to integrate front-end and back-end components seamlessly to create fully functional web applications. They gain experience incommunication between client and server, handling HTTP requests, and implementing RESTful APIs.
4. **Version Control and Collaboration:** Interns become proficient in using version control systems like Git for managing code repositories, collaborating with team members,and tracking changes to their projects effectively.
5. **Problem-Solving Skills:** Interns develop strong problem-solving skills by tackling real-world challenges encountered during project development. They learn to debug code, identify and resolve issues efficiently, and adopt effective troubleshooting strategies.

OUTCOMES:

1. Interns can design and develop responsive, user-friendly front-end interfaces that meetmodern web standards and accessibility guidelines.
2. Interns are capable of developing scalable back-end systems, implementing CRUD (Create, Read, Update, Delete) operations, and integrating databases into their applications securely.
3. Interns are adept at diagnosing and resolving technical issues independently, reducing development time and ensuring the stability and reliability of their applications.

CHAPTER 2: OVERVIEW OF THE ORGANIZATION

2.1 Introduction of the Organization

SmartInternz is a dynamic organization dedicated to bridging the gap between academiaand industry by providing innovative internship programs tailored to the needs of both students and employers. With a focus on fostering practical skills and real-world experience, SmartInternz offersa wide range of internship opportunities in emerging technologies, including data analytics, artificial intelligence, machine learning, software development, and more.

2.2 Vision, Mission, Values of Organization

Vision

SmartInternz envisions a world where every individual has the opportunity to acquire practicalskills and real-world experience to succeed in the rapidly evolving digital landscape. Our visionis to empower the next generation of talent by providing access to high-quality internship programs, mentorship, and resources, fostering innovation and excellence in the global workforce.

Mission

Our mission at SmartInternz is to bridge the gap between education and industry by offering immersive internship experiences that enable students and professionals to gain practical skills,hands-on experience, and industry exposure. We strive to collaborate with leading companies, educational institutions, and mentors to create meaningful learning opportunities that prepare individuals for successful careers in the digital age.

Values

We are committed to upholding the highest standards of quality and excellence in everything we do, from program development to mentorship and support services. We embrace innovationas a driving force for continuous improvement and growth, constantly seeking new ways to enhance our programs and services to meet evolving.

2.3 Policy of Organization In relation to the Intern Role

The policy of SmartInternz organization regarding the intern role emphasizes fostering a supportive and enriching environment that prioritizes the learning and professional development of interns. Interns at SmartInternz are regarded as valued members of the team, entrusted with meaningful responsibilities and opportunities to contribute to real-world projectsunder the guidance of experienced mentors.

The organization is committed to providing interns with hands-on experience, exposure to industry best practices, and access to resources that facilitate their growth and skill development. SmartInternz encourages interns to take initiative, ask questions, and seek feedback to enhance their learning journey and maximize their potential.

Moreover, SmartInternz recognizes the importance of work-life balance and ensures that interns have access to adequate support and resources to manage their workload effectively. The organization values diversity, inclusion, and collaboration, fostering a culture where interns feel respected, supported, and empowered to thrive in their roles.

Overall, SmartInternz's policy regarding the intern role reflects its dedication to nurturing talent, fostering professional growth, and providing a rewarding experience that prepares interns for successful careers in their respective fields.

2.4 Organizational Structure

SmartInternz operates with a dynamic organizational structure designed to facilitate agility, collaboration, and innovation. At the helm is a leadership team comprising visionary executiveswho steer the company's strategic direction and ensure alignment with its mission and values. Functional departments such as Operations, Program Development, Marketing, and Technology work in concert to deliver high-quality internship programs, foster industry partnerships, and provide exceptional support to participants. Within this structure, cross- functional teams collaborate on various initiatives, driving creativity and efficiency Roles and Responsibilities of the Organization, in which the intern is placed

In SmartInternz organization, employees placed to oversee interns hold multifaceted roles and responsibilities aimed at fostering a conducive learning environment while ensuring project success. These employees act as mentors, guiding interns through theirassigned tasks, providing constructive feedback, and facilitating their professional development. They also serve as project leads, coordinating with interns to define project objectives, allocate resources, and monitor progress. Additionally, employees in these roles collaborate with other teammembers to integrate internsseamlessly into project workflows, encourage knowledge sharing, and cultivate a supportive team culture. Overall, they play a pivotal role in empowering interns to learn, grow, and contribute effectively to the organization's objectives.

2.5 Performance of Organization

As of the latest financial reports, SmartInternz has demonstrated impressive performance across key metrics. With a steady increase in turnover and profits, the organizationhas exhibited robust financial health and sustainable growth. SmartInternz has significantly expanded its market reach, establishing a strong presence both locally and internationallythrough strategic partnerships and targeted marketing initiatives. This enhanced market reach has translated into a substantial increase in market value, reflecting investor confidence and the organization's position as a leader in the field of providing immersiveinternship experiences and digital skills development.

2.6 Future Plans of the Organization

SmartInternz is poised for a future marked by continued expansion and innovation, with plans to further solidify its position as a leader in providing immersive internship experiences and career development opportunities. The organization aims to diversify its program offerings, catering to a broader range of industries and skill domains, while leveraging emerging technologies and educational methodologies to enhance the learning experience

CHAPTER 3: INTERNSHIP PART

Description

In SmartInternz's full-stack development internship, participants typically engage in a range of activities and responsibilities designed to provide them with a comprehensive understanding and practical experience in both front-end and back-end web development. Here's an overview of what interns might expect:

Activities/Responsibilities

1. Interns usually start by learning fundamental web development technologies such as HTML, CSS, and JavaScript for front-end development. They may also delve into back-end technologies like Node.js, Express.js, and databases such as MongoDB or SQL.

2. Interns are often assigned hands-on projects to apply the concepts they learn during the training sessions. These projects could involve building web applications from scratch, implementing specific features, or solving real-world problems using the skills acquired.

3. Interns might collaborate with peers on group projects or work in teams to simulate real-world development environments. Collaboration helps them learn how to communicate effectively, manage code repositories using version control systems like Git, and work in an agile development setup.

4. Throughout the internship, interns are likely to encounter various challenges and obstacles in their projects. They're expected to troubleshoot issues independently, seek guidance from mentors or fellow interns when needed, and develop problem-solving skills crucial for a career in software development.

5. Interns may participate in code reviews where they present their work to mentors or senior developers for feedback and suggestions for improvement. This process helps interns refine their coding style, adhere to best practices, and learn from more experienced professionals.

6. The field of web development is dynamic, with new technologies and frameworks emerging regularly. Interns are encouraged to stay updated with industry trends, explore advanced topics, and continuously enhance their skills beyond the core curriculum provided during the internship.

7. Interns often learn about the deployment process, including configuring servers, managing databases, and deploying their applications to hosting platforms such as Heroku, AWS, or Azure. Understanding deployment strategies is essential for launching web applications to production environments.

8. Writing clear and concise documentation is another crucial aspect of software development. Interns may be responsible for documenting their code, project requirements, and deployment procedures to ensure smooth collaboration and future maintenance of the applications they build.

By actively participating in these activities and shouldering these responsibilities, interns can gain valuable hands-on experience and develop the skills necessary to become proficient full-stack developers.

CHAPTER 4: ACTIVITY LOG & WEEKLY REPORT

ACTIVITY LOG FOR THE FIRST WEEK (15.07.2024 to 19.07.2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (15.07.2024)	Introduction to Full Stack	What is Full Stack and structure of internship program	
Day-2 (16.07.2024)	Introduction to HTML Fundamentals	What is HTML, syntax used in HTML and uses of it	
Day–3 (17.07.2024)	HTML fundamentals and elements used	Heading elements, paragraph elements, table elements	
Day–4 (18.07.2024)	CSS Basics	Definiton of CSS, uses of CSS, properties	
Day–5 (19.07.2024)	CSS Layouts, Grid and Flexbox	Layouts used in CSS, Syntax and examples on flexbox features.	

DAY - 1: Introduction to Full Stack:

Full stack development refers to the practice of handling all layers of a web application, from the front-end (client-side) to the back-end (server-side), including database management, API integration, and everything in between.

Internship program:

Week 1-2: Onboarding and Front-end Fundamentals
Week 3-4: Front-end Development
Week 5-6: Back-end Development
Week 7-8: Database Management and Integration
Week 9-10: Project Completion and Deployment

DAY -2: Introduction to Html:

HTML (Hyper Text Markup Language) is a standard markup language used to create web pages. It's the backbone of a website, providing structure and content.

HTML Document Structure:

<!DOCTYPE html>: Declaration of document type.
<html>: Root element.
<head>: Contains metadata (title, styles, scripts).
<body>: Contains visible content.

DAY-3: Html Fundamentals and Elements:

Common Html Elements:

- Headings: <h1>, <h2>, <h3>, etc.
- Paragraphs: <p>
- Links: <a>
- Images: <img>
- Lists: <ul>,

Html Forms:

1. <form>: Form container
2. <input>: Text input
3. <textarea>: Multi-line text input
4. <select>: Dropdown menu
5. <button>: Form submission button

DAY-4: CSS Basics:

Definition of Css: CSS is a styling language used to control the layout and appearance of web pages written in HTML or XML.

CSS Uses: Website design and layout, Responsive web design, Mobile-friendly websites, application design, Progressive web apps (PWAs)

DAY-5:CSS Layouts and Flexbox:

Layouts Display Property: block, inline-block, flex, grid
Layout Models: Block Layout (block, inline),Table Layout (table, table-row, table-cell), Flexbox Layout (flex), Grid Layout (grid)

ACTIVITY LOG FOR THE SECOND WEEK (22.07.2024 to 26.07.2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (22-07-2024)	Introduction to JavaScript	JavaScript syntax,uses and examples	
Day-2 (23-07-2024)	Introduction to ES6 javascript version	Difference between javascript and ES6	
Day–3 (24-07-2024)	Introduction to React.js	React.js components and hands-on session on real world projects	
Day–4 (25-07-2024)	JS syntax discussion	Various syntax has been discussed used in front end	
Day–5 (26-07-2024)	Creating Components	Live practice session on how to create react components	

DAY-1: Introduction to Javascript

JavaScript: JavaScript is a high-level, dynamic, and interpreted programming language used for client-side scripting on the web. It's a crucial tool for creating interactive web pages, web applications, and mobile applications.

Variables and Data Types:

- Variable declaration: let, const, var
- Data types: Number: let x = 5; String: let name = "John"; Boolean: let isAdmin = true;

DAY-2: Introduction TO ES6 Javascript Version:

ES6: ES6, also known as ECMAScript 2015, is the sixth major release of the ECMAScript language standard, which is the foundation of JavaScript.

Difference Between ES6 and Javascript:

- Version: JavaScript is the overall language, while ES6 is a specific version.
- Syntax: ES6 introduced new syntax features like let, const, arrow functions, and template literals.
- Features: ES6 added features like classes, modules, promises, and symbols.

DAY-3: Introduction to React.js:

React.js: React is a JavaScript library for building user interfaces. It's used for creating reusable UI components and managing the state of complex applications.

Components: In React, a component is a reusable piece of code that represents a UI element. Components can contain other components, making it easy to build complex interfaces.

Types of Components:

- Functional Components: Simple, stateless components that accept props and return JSX.
- Class Components: Stateful components that extend the React.Component class.
- Pure Components: Optimized versions of class components that minimize re-renders.

DAY-4: JS Syntax Discussion:

Variables and Data Types: Declaring variables: let, const, var, Basic data types: number, string, boolean, null, undefined, Complex data types: array, object, function

Operators: Arithmetic operators: +, -, *, /, %, ** Logical operators: &&, ||, !

Assignment operators: =, +=, -=, *=, /=, %=

Functions: 1.Function declaration: function name() { ... }

2.Function expression: let name = function() { ... };

3.Arrow function: let name = () => { ... };

DAY-5: Creating Components:

Open the React project in your code editor, Create a new file called HelloWorld.js in the src directory, Import React and create a simple component:

```
import React from 'react';
function HelloWorld() {
  return <h1>Hello, World!</h1>;}
export default HelloWorld;
```

ACTIVITY LOG FOR THE THIRD WEEK (29.07.2024 to 02.08.2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (29-07-2024)	Forms in React	Structure of forms and purpose of forms, fields it should contain	
Day-2 (30-07-2024)	React Lifecycle and Hooks	Constructor, render and component DidMount methods	
Day–3 (31-07-2024)	Introduction to Node.js	Node.js uses javascript on the server and open source environment	
Day–4 (01-08-2024)	Setting up Node.js in VS Code and Intergrated developed environment	Run commands to start backend of application	
Day–5 (02-08-2024)	Introduction to Core module in Node.js	Http module,Url module creation of own modules	

DAY-1: Forms in React:

Structure of Forms:

- Form Fields: Input fields, checkboxes, radio buttons, select dropdowns, etc.
- Labels: Text describing each form field.
- Error Messages: Displayed when validation fails.
- Submit Button: Triggers form submission.

Fields in Forms: Text Input (e.g., name, email, password), Checkboxes (e.g., terms and conditions), Radio Buttons (e.g., gender), Select Dropdowns (e.g., country)

DAY-2: **React Lifecycles and Hooks:**

Constructor Method:

- Called when component is initialized.
- Used to:
- Initialize state.
- Bind event handlers.

DAY-3: Introduction to Node.Js:

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine that allows developers to run JavaScript on the server-side. It provides an event-driven, non-blocking I/O model, making it lightweight and efficient.

DAY-4: Setting Up Nodejs in VS Code and Integrated Developed Environment:

(IDE): 1. Install Node.js

2. Install VS Code
3. Install Node.js Extension in VS Code
4. Create a New Node.js Project
5. Initialize a New Node.js Project
6. Create a New File for Your Node.js Application

DAY-5: Introduction to Core Module in Node.js:

Http Module: The Http module in Node.js allows you to create a web server and handle HTTP requests.

Key Features: Creating an HTTP server, HTTP requests and responses, Supporting HTTP methods (GET, POST, PUT, DELETE, etc.)

Url Module: The Url module in Node.js provides a way to parse and manipulate URLs.

Key Features: Parsing URLs into components (protocol, hostname, pathname, etc.), Manipulating URL components, Creating new URLs

Creation of Own Modules: In Node.js, you can create your own modules using JavaScript files.

Key Features: Exporting functions and variables, Importing modules using require()

ACTIVITY LOG FOR THE FORTH WEEK (05-08-2024 to 09-08-2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (05-08-2024)	Introduction to Express.js	Expressjs is framework work on node web server functionality.	
Day-2 (06-08-2024)	Express js setup	Steps to follow to setup the express js environment	
Day–3 (07-08-2024)	Building and Principles of Rest API'S	It is used to navigate between various url's	
Day–4 (08-08-2024)	CRUD operations with rest	Operartions like create, Read,update,delete to Manipulate data.	
Day–5 (09-08-2024)	Introducgion to Middleware and Rest api's in express	Middleware services used to transfer data between url's	

DAY-1: Introduction to Express.js:

Express.js is a popular Node.js web framework that provides a flexible and efficient way to build web applications. It extends the Node.js HTTP module, adding support for routing.

Express.js Core Components:

- Application: The main Express.js instance.
- Router: Handles routing for the application.
- Request: Represents an incoming HTTP request.
- Response: Represents the outgoing HTTP response.
- Middleware: Functions that perform tasks before or after request handling.

Express.js Middleware: Body Parser: Parse request bodies (e.g., JSON, URL-encoded), Cookie Parser: Parse cookies, Session: Manage user sessions, Authentication: Handle user authentication.

DAY-2: Express Is Setup:

Here are the steps to set up an Express.js environment:

Step 1: Install Node.js

Step 2: Install npm (Node Package Manager)

Step 3: Create a New Project Folder

Step 4: Initialize a New npm Project

Step 5: Install Express.js

Step 6: Create a New Express.js App

Step 7: Run Your Express.js App

Step 8: Install Additional Dependencies (Optional)

Step 9: Configure Your Environment (Optional)

Step 10: Test and Deploy Your App

DAY-3: Building and Principles of REST API'S:

REST (Representational State of Resource) API is an architectural style for designing networked applications. It's based on the idea of resources, which are identified by URIs, and can be manipulated using a fixed set of operations.

Key Principles of REST API:

- Resource-based: Everything in REST is a resource.
- Client-Server Architecture: Client and server are separate.
- Stateless: Server doesn't maintain client state.
- Cacheable: Responses can be cached.

HTTP Methods in REST API:

- GET: Retrieve a resource.
- POST: Create a new resource.
- PUT: Update an existing resource.

- DELETE: Delete a resource.

API Navigation:

- URL Parameters: /users/{id}
- Query Parameters: /users?name=John&age=30
- Path Parameters: /users/{id}/orders/{orderId}

DAY-4: CRUD Operations with Rest:

CRUD (Create, Read, Update, Delete) operations are essential for manipulating data in a RESTful API. Here's a breakdown of each operation and how they correspond to REST methods:

Create:	1. REST Method: POST
POST/users	2 Description: Create a new resource
	3. Example: Create a new user
Read:	1. REST Method: GET
GET /users	2. Description: Retrieve a resource or a collection of resources
	3. Example: Get all users
Update:	1. REST Method: PUT/PATCH
PUT /users/1	2. Description: Update an existing resource
	3.Example: Update user with ID 1
Delete	1. REST Method: DELETE
PUT /users/1	2. Description: Delete a resource
	3.Example: Delete user with ID 1

DAY-5:

Middleware functions are used to perform specific tasks before or after processing a request. They can:

- Authenticate requests
- Validate data
- Parse request bodies
- Log requests
- Handle errors

Types of Middleware:

- Application-level middleware
- Router-level middleware
- Built-in middleware (e.g., express.json(), express.urlencoded())
- Third-party middleware (e.g., body-parser, cors)

REST APIs in Express: REST (Representational State of Resource) APIs follow an architectural style for designing networked applications.

ACTIVITY LOG FOR THE FIFTH WEEK (12-08-2024 to 17-08-2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (12-08-2024)	Overview of frontend, backend and Mongodb	Technologies covered in overall development of web-application	
Day-2 (13-08-2024)	Implementation of Authorization, Authentication with Mongodb	Tokens used to ensure security of web application	
Day–3 (14-08-2024)	Index Types and working with Object ID	Indexing of data in database and retrieve id's	
Day–4 (16-08-2024)	Basic Query operations	Queries in mongodb to retrieve data from database	
Day–5 (17-08-2024)	Aggregation on frameworks	Mathematical operations to manipulate the datasets	

DAY-1: Overview of Frontend-Backend and Mongodb:

Here's an overview of frontend, backend, and MongoDB technologies typically used in web application development:

Frontend Technologies: HTML (Hypertext Markup Language). CSS (Cascading Style Sheets), JavaScript, React.js, Bootstrap, Material-UI

Backend Technologies: Node.js, Express.js, RESTful APIs

Database Technologies: MongoDB

MongoDB Specific Technologies: MongoDB Compass (GUI tool)

Development Tools: Visual Studio Code, GitHub, npm (Node Package Manager)

Security Considerations:

- Authentication (OAuth, JWT)
- Authorization (Role-based access control)
- Encryption (HTTPS, SSL/TLS)
- Input Validation
- Error Handling

DAY-2: Implementation of Authorization Authentication with Mongodb:

Authentication:

1. User Registration: Store user credentials (username, password) in MongoDB.
2. Password Hashing: Use bcrypt or similar library to hash passwords.
3. Login: Verify user credentials and generate token.

Authorization:

1. Token-based Authorization: Use tokens to authenticate requests.
2. Role-based Access Control (RBAC): Assign roles to users (e.g., admin, user).
3. Permission-based Access Control: Define permissions for each role.

DAY-3: Index Types and Working with Object Id:

Index Types: Single Field Index: Indexes a single field, Compound Index: Indexes multiple fields, Multi-Key Index: Indexes arrays, Text Index: Indexes text data. Hashed Index: Indexes hashed values.

Object ID Indexing:

1. ObjectId: Unique identifier for documents.
2. _id field: Automatically created by MongoDB.

Retrieving IDs: Find by ID: db.collection.findById(id), Find by Multiple IDs: db.collection.find({ _id: { $in: [id1, id2, id3] } }), Get IDs only: db.collection.find({}, { _id: 1 })

DAY-4: Basic Query Operations:

Query Methods: find(): Retrieve all documents matching the query, findOne(): Retrieve one document matching the query, aggregate(): Perform aggregation operations, count(): Count documents matching the query.

DAY-5: Aggregation on Framework:

The Aggregation Framework provides a powerful way to perform mathematical operations on datasets

Common Aggregation Operations: sum: Calculates the sum of a field, $avg: Calculates the average of a field, $max: Finds the maximum value of a field.

ACTIVITY LOG FOR THE SIXTH WEEK (19-08-2024 to 23-08-2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (19-08-2024)	Aggregation pipeline	Group the data together into dataset	
Day-2 (20-08-2024)	Introduction to Mongoose	Library in mongodb to restart the server	
Day–3 (21-08-2024)	Hands on practice on crud operations and mongoose	Creating a database and performing crud operations on database	
Day–4 (22-08-2024)	Connecting the frontend and backend	Commands to connect the user interface and server	
Day–5 (23-08-2024)	Figma tools	UI/UX designer tools used for developing layout for application	

DAY-1: Aggregation Pipeline:

The Aggregation Pipeline consists of stages that process data in a pipeline fashion.
Grouping Data: $group: Groups documents by a field.
Grouping Operators:

- $avg: Calculates the average of a field.
- $sum: Calculates the sum of a field.
- $max: Finds the maximum value of a field.
- $min: Finds the minimum value of a field.
- $count: Counts the number of documents.

DAY-2: Mongoose

Mongoose: Mongoose is a popular Object Data Modeling (ODM) library for MongoDB.
Features: 1. Simplifies MongoDB interactions
2. Provides schema-based data modeling
3. Supports validation, hooks, and plugins
4. Improves data consistency and security

Restarting Server: To restart the server, use the following commands:

1. npm stop
2. npm start
3. nodemon restart

DAY-3: Hands-On Practice on CRUD Operations and Mongoose:

Hands-on Practice on CRUD Operations and Mongoose:
Step 1: Create a Database
Step 2: Install Required Packages
Step 3: Create Mongoose Model
Step 4: Create Express Server
Step 5: Perform CRUD Operations

DAY-4: Connecting Frontend and Backend:

Connecting Frontend and Backend:

Step 1: Install required packages

Step 2: Set up API endpoints

Step 3: Create API requests

Step 4: Send requests from frontend to backend

DAY-5: Figma Tools:

Figma Tools: Frames: Create and manage design layouts, Layers: Organize and structure design elements, Shapes: Draw and manipulate vector shapes, Text: Style and format text, Icons: Use pre-made icons or create custom ones, Components: Reusable design elements, Prototyping: Create interactive prototypes, Design Systems: Manage and scale design systems

ACTIVITY LOG FOR THE SEVEN WEEK (02-09-2024 to 06-09-2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (02-09-2024)	Knowledge sessions on Github	Introduction to github, uses and principles	
Day-2 (03-09-2024)	Projects	Explaination of projects titles with examples	
Day–3 (04-09-2024)	Full-stack development MERN use cases	Use cases for development of projects	
Day–4 (05-09-2024)	Doubts session on frontend, backend, databases technologies	Brief revision on topics discovered and gentle preparation for further topics	
Day–5 (06-09-2024)	Project Orientation and Mentoring Session	Guidelines for development of projects	

DAY-1: Knowledge Sessions on GitHub:

Introduction to GitHub: GitHub is a web-based platform for version control and collaboration software development projects.

Uses of GitHub:

- Open-source software development
- Collaborative coding
- Project management
- Code review and feedback
- Version control

Principles of GitHub: Forking: Create a copy of a repository

- Cloning: Download a repository to local machine
- Committing: Save changes to local repository
- Pushing: Upload changes to remote repository
- Pulling: Download changes from remote repository

DAY-2: Web Development Projects:

E-commerce Website- Example: Building an online shopping platform using React, Node.js, and MongoDB.

Social Media Platform -Example: Creating a social networking site using Python, Django, and MySQL.

Personal Portfolio Website -Example: Designing and developing a personal website using HTML, CSS, and JavaScript.

DAY-3: Full Stack Development MERN UseCases:

MERN Use Cases: Social Media Platforms, E-commerce Websites, Blogging Platforms, Online Forums, Project Management Tools, IOT Applications

DAY-4: Doubts Session on Frontend Backend and Database Technologies:

Common Doubts and Questions: 1. How to choose the right frontend framework?
2. What's the difference between RESTful APIs and GraphQL?
3. How to optimize database performance?
4. What's the role of Node.js in backend development?

DAY-5 Project Orientation and Mentoring Session:

Project Development Guidelines: 1. Define project scope and objectives.
2. Conduct market research and analysis.
3. Create a detailed project plan.

ACTIVITY LOG FOR THE EIGTH WEEK (09-09-2024 to 13-09-2024)

Day &Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (09-09-2024)	Installation on Tailwind CSS	CSS components available that can be inherited into project	
Day-2 (10-09-2024)	Installation on Nodejs	To setup the back	
Day–3 (11-09-2024)	Hands on practice on	Group the data together into dataset	
Day–4 (12-09-2024)	Doubts session on frontend, backend, databases technologies	Brief revision on topics discovered and gentle preparation	
Day–5 (13-09-2024)	Overview of frontend, backend and Mongodb	Technologies covered in overall development of web-application	

DAY-1: Installation on Tailwind Css:

Step 1: Install Tailwind CSS using npm or bash npm install tailwindcss

Step 2: Import Tailwind CSS into your CSS file

Node.js: Installation of Node.js:

Step 1: Download and Install Node.js

- Go to the Node.js website: (link unavailable)
- Click on the "Download" button.
- Select the correct operating system (Windows, macOS, or Linux).
- Choose the desired version (LTS or Current).
- Follow the installation instructions.

DAY-2: SetUp Backend: Node.js Resources:

- Node.js Official Documentation
- Node.js Tutorials

DAY-3: Hands on Practice On:

Common Grouping Techniques:

- Aggregate Functions: SUM, COUNT, AVG, MAX, MIN.
- Group By: SQL clause for grouping data.
- Pivot Tables: Spreadsheets for grouping and analyzing data.

DAY-4: Doubts Session on Frontend Backend Data base:

Revision Topics:

- Frontend-backend interaction
- Database schema design
- API design principles
- Security best practices
- Deployment strategies

DAY-5: Overview on Frontend Backend and Mongodb:

Technologies Covered in Overall Development:

Frontend Technologies: HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), JavaScript, React.js, Bootstrap, Material-UI

Backend Technologies: Node.js, Express.js, RESTful APIs

Database Technologies: MongoDB

MongoDB Specific Technologies: MongoDB Compass (GUI tool)

ACTIVITY LOG FOR THE NINETH WEEK (16-09-2024 to 20-09-2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (16-09-2024)	Hands on practice on UI components	Developing the front-end landing page	
Day-2 (17-09-2024)	Doubts session on frontend, backend, databases technologies	Brief revision on topics discovered and gentle preparation	
Day–3 (18-09-2024)	Flex tailwind and CSS projects	CSS components available that can be inherited into project	
Day–4 (19-09-2024)	Projects and use cases	Use cases for development of projects	
Day–5 (20-09-2024)	Doubts session on frontend, backend, databases technologies	Brief revision on topics discovered and gentle preparation	

DAY-1: Hands on UI Components:

Create a responsive landing page using React-BootstrapCSS, and JavaScript. Incorporate the following UI components:

- Navigation Bar
- Background Image
- Features Section
- Call-to-Action (CTA) Button
- Footer

DAY-2: Doubts Session on Frontend Backend Database Technologies:

Revision Topics:

- Frontend
- Backend
- Databases

DAY-3: Tailwind and Css Projects:

Tailwind CSS Projects:

- Tailwind UI
- Tailwind CSS Components
- Tailwind CSS Templates

DAY-4: Projects and UseCases:

Use Cases:

- User authentication and authorization
- Data visualization and reporting
- Real-time data processing
- Integration with third-party APIs
- Payment gateway integration

DAY-5: Doubts Session:

Revision Topics:

- Frontend
- Backend
- Databases

ACTIVITY LOG FOR THE TENTH WEEK (23-09-2024 to 27-09-2024)

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day–1 (23-09-2024)	Explanation of projects	Explaination of projects titles with examples	
Day-2 (24-09-2024)	Doubts session on frontend, backend, databases technologies	Brief revision on topics discovered and gentle preparation	
Day–3 (25-09-2024)	HTML fundamentals and elements used	Heading elements, paragraph elements, table elements	
Day–4 (26-09-2024)	Doubts session on frontend, backend, databases technologies	Brief revision on topics discovered and gentle preparation	
Day–5 (27-09-2024)	Index Types and working with Object ID	Indexing of data in database and retrieve id’s	

DAY-1: Explanation of Projects:

Web Development Projects

DAY-2: Doubts Session:

Revision Topics:

- Frontend
- Backend
- Databases

DAY-3: Html Fundamentals and Elements Used:

HTML Fundamentals and Elements:

- Heading Element
- Paragraph Element
- Table Element
- Other Essential Element
- Semantic Element
- Accessibility Element

DAY-4: Doubts Seesion:

Revision Topics:

- Frontend
- Backend
- Databases

DAY-5: Index Types and Working With Object ID:

Index Types:

- Single Field Index: Indexes a single field.
- Compound Index: Indexes multiple fields.
- Multi-Key Index: Indexes arrays or embedded documents.
- Text Index: Indexes text data for text search.

Object ID Indexing:

- _id field: Automatically indexed primary key.
- ObjectID(): Creates a new ObjectID.
- ObjectId("..."): Creates an ObjectID from a string

CHAPTER 5: OUTCOMES DESCRIPTION

By achieving these learning objectives and outcomes, interns in the SmartInternz full-stack development internship program acquire a strong foundation in web development and are well-prepared to pursue career opportunities as full-stack developers in the industry.

- Interns are equipped with the mindset and skills necessary to thrive in a dynamic and rapidly changing technology landscape, enabling them to pursue lifelong learning and career advancement in web development.
- Interns can effectively manage project timelines, prioritize tasks, and deploy their web applications to production environments, ensuring smooth and efficient project delivery.
- Interns are adept at diagnosing and resolving technical issues independently, reducing development time and ensuring the stability and reliability of their applications.
- Interns can work collaboratively on projects, contribute to shared codebases, resolve merge conflicts, and maintain a structured development workflow using Git branching strategies.
- Interns can build end-to-end web solutions, connecting user interfaces with server-side logic, and implementing features such as user authentication, data validation, and session management.
- Interns are capable of developing scalable back-end systems, implementing CRUD (Create, Read, Update, Delete) operations, and integrating databases into their applications securely.
- Writing clear and concise documentation is another crucial aspect of software development. Interns may be responsible for documenting their code, project requirements, and deployment procedures to ensure smooth collaboration and future maintenance of the applications they build.