

INTERNSHIP REPORT

A report submitted in partial fulfillment of the requirements for the Award of Degree of

**BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE & ENGINEERING
By
Varenyam Singh Baghel
Under Supervision of:
Dr. Rajneesh Kumar Patel
(Duration: AUG 2023 to NOV 2023)**



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
VIT BHOPAL UNIVERSITY
APRIL - 2025**

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING VIT

BHOPAL UNIVERSITY



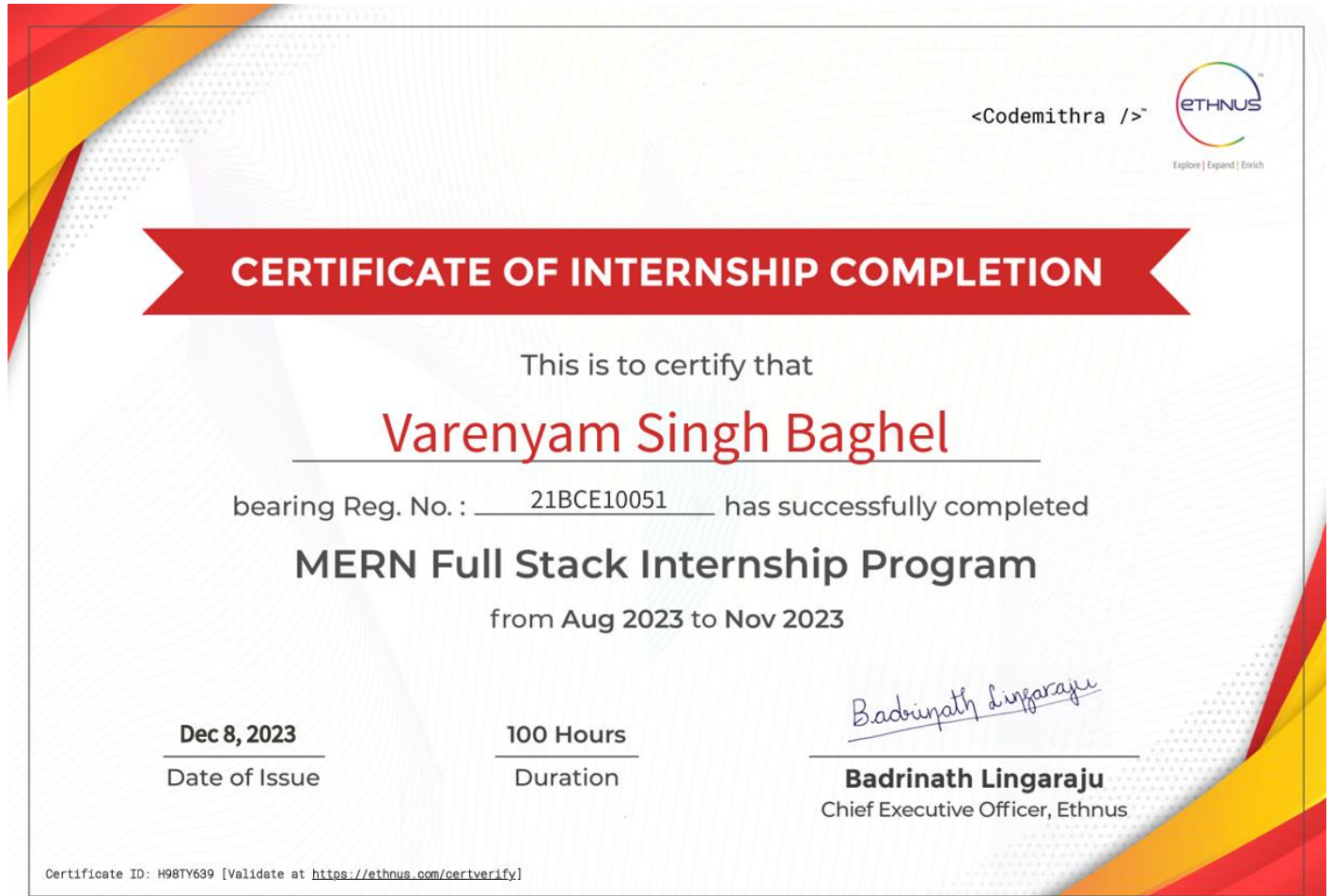
VIT[®]
B H O P A L
www.vitbhopal.ac.in

CERTIFICATE

This is to certify that the “**Internship report**” submitted by **Varenyam Singh Baghel (Reg. No.: 21BCE10051)** is work done by him and submitted during 2023-24 academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING**.

College Internship Guide
Dr. Rajneesh Kumar Patel
Supervisor

GLOBAL CERTIFICATE



Validate at: <https://ethnus.com/certverify>

Certificate ID: H98TY639

ACKNOWLEDGEMENT

First, I would like to thank **Mr. Badrinath Lingaraju**, CEO of Ethnus, for providing me with the opportunity to pursue and complete certifications under their esteemed platform.

I also would like all the people that worked along with me at Ethnus projects with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense gratitude that I acknowledge the help and guidance of these individuals during my certification journey.

I am highly indebted to Dean **Dr. Pushpinder Singh Patheja** for the facilities and encouragement provided to successfully complete the certification programs.

I would like to thank my Program Chair, **Dr. Dheresh Soni**, for his valuable insights and constructive suggestions throughout this academic endeavor.

I extend my sincere thanks to **Dr. Vivek Sharma** for his support and guidance in choosing and completing the certifications from the above-mentioned organization.

I would like to thank **Dr. Rajneesh Kumar Patel Sir** for his support and advices to get and complete internship certification in above said organization.

I am also extremely grateful to my department staff members and friends who supported and motivated me throughout the process and contributed to the successful completion of the certifications.

Varenyam Singh

21BCE10051

ABSTRACT

The Ethnus MERN Stack Internship Program, has significantly enhanced my expertise in full stack web development by providing both theoretical knowledge and hands-on experience.

Ethnus

Ethnus is an IT training and consulting firm specializing in upskilling students and professionals in various technical domains. The **Ethnus MERN Stack Internship Program (Global Certification)** is designed to equip learners with full-stack development skills through live classes and hands-on projects. The training not only provides essential knowledge in building and managing modern web applications but also focuses on developing real-world, market-ready software solutions that businesses can adopt.

The Problem or Opportunity

Cloud computing is revolutionizing industries by offering cost-efficient, flexible, and scalable IT solutions. Full-stack development is in high demand, with companies seeking skilled developers proficient in building and managing web applications. Ethnus addressed this gap by offering a structured MERN Stack training program, ensuring participants gained practical experience by working on real-world projects. Our team specifically worked on a **Personal Finance Tracker**, while other teams were assigned different projects.

Methodology

This internship was structured to provide a comprehensive, hands-on learning experience using modern web technologies. The methodology included:

- **Hands-on Training** – Live coding sessions covering MongoDB, Express.js, React.js, and Node.js to develop expertise in full-stack development.
- **Project-Based Learning** – Developing real-world applications using agile methodologies, ensuring iterative improvements and collaboration.
- **Code Reviews & Debugging** – Implementing best practices for writing clean, maintainable code through peer and mentor feedback.
- **API Development** – Building robust RESTful APIs with Express.js and integrating them with front-end applications for seamless data flow.
- **State Management** – Using Redux, Context API, and React Hooks to efficiently manage application state and improve performance.
- **Authentication & Security** – Implementing OAuth, JWT, and session-based authentication to ensure secure user access and data protection.
- **Cloud Deployment** – Hosting applications using Render, AWS, and Vercel, providing real-world experience in deploying scalable web applications.

While other teams were assigned different projects, our team specifically developed a Personal Finance Tracker, implementing the above methodologies to ensure a well-structured and functional project. The project was successfully deployed on Render, demonstrating our ability to manage cloud-based hosting and deployment strategies.

Key Parts of the Report & Findings

The report highlights the structured learning paths for both certifications, demonstrating the significance of

cloud computing and full-stack development. The Ethnus MERN Full Stack Internship Program emphasized hands-on experience by engaging participants in building a Personal Finance Tracker. The project followed a full-stack development approach, integrating Node.js, Express.js, MongoDB, and EJS templates to create a functional web application. The backend, implemented with Express.js, handles API requests efficiently, while the MongoDB database stores customer and inventory data securely. The frontend, using EJS and CSS, delivers an interactive user interface.

To enhance security and user authentication, React.js was integrated to provide a seamless login and registration system. JWT (JSON Web Token) authentication was implemented to secure user sessions, ensuring authorized access to inventory management features. React components were used to handle authentication states dynamically, improving the user experience with real-time validation and feedback mechanisms.

Deployment played a crucial role in ensuring the application was accessible and functional in a cloud environment. The Personal Finance Tracker was deployed on Render, enabling seamless accessibility and real-time updates. The deployment process involved configuring the backend on a secure server, setting up a MongoDB Atlas database for scalability, and ensuring that the frontend remained responsive across different devices. This experience provided practical knowledge of hosting web applications and maintaining cloud-based infrastructures efficiently.

Key findings indicate that this practical experience enhanced participants' problem-solving skills, improved their understanding of database interactions, and strengthened their ability to develop scalable web applications. Overall, the report underscores how these certifications and projects contribute to industry readiness.

Benefits to the Company/Institution

The AWS certification program enables professionals and businesses to adopt cloud solutions efficiently, allowing organizations to hire certified experts who optimize cloud usage and strengthen security. Ethnus MERN Full Stack Internship Program contributes to skill development, bridging the gap between academic learning and industry expectations. The Personal Finance Tracker is a useful tool for managing income, expenses, savings, and investments efficiently. It helps users track their spending habits, set budgets, and achieve financial goals. By providing insights into cash flow and categorizing expenses, it promotes better money management and prevents overspending. It can be built as a web or mobile app with features like expense logging, budget analysis, and financial reports. Integration with bank accounts and visualization tools (charts/graphs) enhances usability.

INDEX

S. no.	Title	Page
1	Learning Objectives	1
2	Weekly Overview Of Internship Activities	2
3	Introduction	5
4	History Of Web Development	6
7	Definition Of Mern Stack	7
8	Ethnus Mern Stack Project Overview - Personal Finance Tracker	10
9	Project Architecture	11
10	Objectives Achieved	14
11	Summary Of Skills Learnt	18
12	Challenges Faced	19
13	Results And Observations	21
14	Conclusion	22
15	Bibliography	23

LEARNING OBJECTIVES

- **Learn Full-Stack Web Development** – Gain hands-on experience with MongoDB, Express.js, React.js, and Node.js (MERN Stack) to build modern web applications.
- **Mastering MERN Stack Technologies:** Gain proficiency in MongoDB, Express.js, React.js, and Node.js, understanding their individual functionalities and how they integrate to form a full-stack development environment.
- **Front-End Development with React.js:** Develop skills in building interactive user interfaces using React.js, including component-based architecture, state management, and integration with external libraries and APIs.
- **Server-Side Development with Node.js:** Learn server-side development concepts with Node.js, including asynchronous programming, event-driven architecture, and working with HTTP requests and responses.
- **Full-Stack Application Development:** Combine the skills learned in database management, AP development, and front-end and back-end technologies to build full-stack web applications from scratch.
- **Problem-Solving and Debugging Skills:** Develop critical thinking and problem-solving skills by tackling real-world challenges and debugging issues encountered during the development process.
- **Effective Communication and Team Collaboration:** Enhance communication skills and learn the importance of effective collaboration within a development team, including task delegation, peer code reviews, and project management methodologies.
- **Project Development:** Build a portfolio of projects showcasing the skills acquired during the internship, including both individual and group projects developed throughout the program.
- **Version Control and Collaboration:** Gain experience with version control systems like Git and collaboration platforms like GitHub, understanding the importance of collaborative development workflows and code management practices.
- **Database Design and Management:** Learn database modeling principles and best practices for MongoDB, including schema design, indexing, and data querying.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

ETHNUS MERN Full Stack Internship Orientation –

21/08/23

WEEK 1

DATE	DAY	MODULE COMPLETED
28/08/23	MONDAY	Started with HTML
01/09/23	FRIDAY	HTML Forms and Tables

WEEK 2

DATE	DAY	MODULE COMPLETED
04/09/23	MONDAY	Started CSS
06/09/23	WEDNESDAY	CSS and JavaScript Introduction for dynamic behavior
08/09/23	FRIDAY	JavaScript maps, filter, reduce, array and object methods

WEEK 3

DATE	DAY	MODULE COMPLETED
11/09/23	MONDAY	JavaScript promises, arrow functions and destructuring
13/09/23	WEDNESDAY	Node.js introduction, basics of HTTP server
15/09/23	FRIDAY	Express.js server and routes

WEEK 4

DATE	DAY	MODULE COMPLETED
18/09/23	MONDAY	Rest API and CRUD operations
20/09/23	WEDNESDAY	Rest API continued
22/09/23	FRIDAY	MongoDB installation and introduction

WEEK 5

DATE	DAY	MODULE COMPLETED
25/09/23	MONDAY	MongoDB Schema and model
27/09/23	WEDNESDAY	Mongoose

WEEK 6

DATE	DAY	MODULE COMPLETED
02/10/23	MONDAY	Node.js Authentication and Authorisation
04/10/23	WEDNESDAY	Session cookies and tokens in node.js
06/10/23	FRIDAY	JWT Token authentication

WEEK 7

DATE	DAY	MODULE COMPLETED
09/10/23	MONDAY	Started with React.js
11/10/23	WEDNESDAY	React hooks
13/10/23	FRIDAY	React router

WEEK 8

DATE	DAY	MODULE COMPLETED
16/10/23	MONDAY	React forms and validation
18/10/23	WEDNESDAY	Introduction to redux
20/10/23	FRIDAY	Redux-promise and Redux-saga

WEEK 9

DATE	DAY	MODULE COMPLETED
23/10/23	MONDAY	Redux-Toolkit
25/10/23	WEDNESDAY	Connecting react with Express API
27/10/23	FRIDAY	Deployment strategies, allotment of project work – Personal Finance Tracker

Demonstration and acceptance of project Personal Finance Tracker – 22/11/23

INTRODUCTION

Overview of the Internship

Internships serve as a bridge between academic learning and real-world applications, providing hands-on experience and industry exposure.

Ethnus MERN Stack Internship Program (Global Certification) – A full-stack web development program focused on MongoDB, Express.js, React.js, and Node.js, culminating in the development and deployment of a real-world project.

The certification has played a crucial role in my professional development, strengthening my technical capabilities and problem-solving skills. This report details my experiences, learnings, and contributions throughout the internship, with a specific focus on the **Personal Finance Tracker**—a full-stack web application that my team developed and deployed.

Purpose and Objectives

The primary objective of this internship was to gain practical experience in cloud computing and full-stack web development through hands-on projects and structured training. The Ethnus MERN Stack Internship provided intensive training in full-stack development, agile methodologies, authentication, API development, and cloud deployment.

Additionally, this internship aimed to:

- Provide in-depth exposure to AWS cloud solutions and their practical applications in modern businesses.
- Develop real-world web applications using the MERN stack, with an emphasis on best coding practices.
- Strengthen API development skills through the creation of RESTful APIs with Express.js.
- Implement secure authentication mechanisms using JWT, OAuth, and session-based authentication.
- Deploy web applications on Render, and Vercel to ensure cloud-based hosting proficiency.
- Enhance problem-solving and debugging skills through rigorous code reviews and debugging sessions.
- Improve project management and collaboration through Agile methodologies, sprint planning, and iterative development cycles.

By the end of this internship, I had gained industry-relevant experience, built a market-ready software product, and significantly enhanced my resume and employability.

Description of the Internship Structure

Ethnus MERN Full Stack Internship Program

- Live coding sessions and training in MERN stack technologies.
- Project-based learning, where teams were assigned different real-world projects.
- Development of a sellable Personal Finance Tracker, showcasing full-stack development skills.
- Integration of React.js authentication (OAuth, JWT, session-based security).
- Deployment on Render, with MongoDB Atlas as the database backend.
- Team collaboration using Agile methodologies, with sprint-based progress tracking.

HISTORY OF WEB DEVELOPMENT

The Early Days (1990s)

- 1990-1991: Tim Berners-Lee invents the World Wide Web, developing HTML, HTTP, and the first web browser.
- 1993: Mosaic, the first graphical web browser, is released, making the internet accessible to the public.
- 1995: JavaScript is introduced, enabling interactivity on websites.
- 1996: CSS (Cascading Style Sheets) enhances web design.
- 1998: Google is founded, revolutionizing online search.

The Rise of Web Applications (2000s)

- 2000: AJAX enables dynamic, fast-loading websites.
- 2003: WordPress launches, simplifying website creation.
- 2004: Web 2.0 emerges, promoting social media and user-generated content.
- 2007: The iPhone drives the need for responsive web design.
- 2008: Google Chrome is introduced, enhancing web performance.

The Modern Era (2010s - Present)

- 2010: HTML5 enables multimedia without plugins.
- 2013: React.js revolutionizes frontend development.
- 2015: ES6 modernizes JavaScript.
- 2017: Progressive Web Apps (PWAs) bridge the gap between websites and mobile apps.
- 2020: WebAssembly (WASM) gains traction, enabling near-native performance for web applications.
- 2023: AI-powered web development tools, like GitHub Copilot and AI-assisted design, streamline coding and design processes.

DEFINITION OF MERN STACK

The MERN stack is a popular JavaScript-based technology stack used for developing full-stack web applications. It consists of four key technologies: MongoDB, Express.js, React, and Node.js, enabling developers to build scalable and efficient applications with a unified language, JavaScript.

Components of MERN Stack

1. **MongoDB** – A NoSQL database that stores data in flexible, JSON-like documents. It provides high scalability, easy data retrieval, and integration with JavaScript-based applications.
 2. **Express.js** – A lightweight and fast web framework for Node.js, simplifying backend development by providing powerful routing, middleware, and HTTP request handling.
 3. **React** – A front-end JavaScript library developed by Facebook, used for building interactive user interfaces with a component-based architecture and efficient rendering via the virtual DOM.
 4. **Node.js** – A runtime environment that allows JavaScript to be executed on the server side, enabling seamless full-stack development and handling asynchronous operations efficiently.
- ### Working of MERN Stack

The MERN stack follows a **three-tier architecture**:

Frontend (React):

In web development, the front-end, as known as client, refers to the part of a web application where all the direct interactions happen between the user and the application.

React, is one of the most popular JavaScript libraries in the world. Introduced by Facebook (now known as Meta) in 2013, it has been widely used by web developers all over the world. It is an open-source project as a public repository on GitHub and is constantly maintained and updated by a community of developers.

React utilized the concept of component-based design, which means that the user interface is divided into discrete, reusable modules known as components. Every component has a unique state that controls how it renders and functions. Moreover, each component can be passed in data, or "props," from its parent component.

Backend (Express.js & Node.js):

Manages API requests, authentication, and business logic.

NodeJs:

NodeJS is an open-source, cross-platform, runtime environment built on the Chrome V8 JavaScript engine. It allows developers to run JavaScript code without the need of a web browser, with high compatibility to various operating systems like Windows, Linux or macOS.

Web servers, web apps, command-line tools, and other types of network applications can be created with NodeJS. Building scalable and real-time applications is suitable to NodeJS thanks to its event-driven, non-blocking input/output style.

NodeJS comes with Node Package Manager (npm), which grants users access to a massive ecosystem of modules and packages.

```

const http = require('http');

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, world!');
});

server.listen(3000, () => {
  console.log('Server listening on port 3000');
});

```

Fig 1: Simple Web server built on NodeJs

ExpressJs:

ExpressJS is a NodeJS web application framework which is lightweight, versatile, and offers a set of robust features for both web and mobile apps. By offering a thin layer of basic web application functionalities, such as routing, middleware, and HTTP handling, it is intended to make it easier than using vanilla NodeJS, to manage and arrange routes in complicated web applications.

ExpressJS is especially well-suited for the MERN stack because it provides developers the ability to quickly build RESTful APIs and handle HTTP requests from the client, such as the GET, POST, PATCH, DELETE method, which are the key features in web applications

Development

Database (MongoDB):

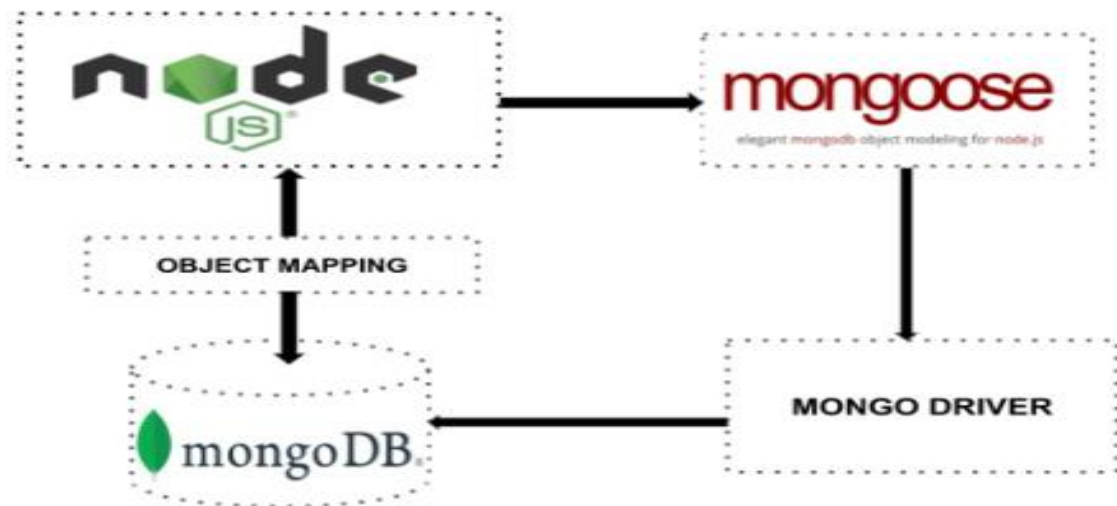
Stores and retrieves data efficiently.

In a typical MERN application, React interacts with Express.js through API calls, which then communicates with MongoDB to fetch or update data, all running on a Node.js server.

MongoDB is a document-oriented database, a type of NoSQL database, which is designed for scalability, high performance, and flexibility. MongoDB is a suitable choice for applications with fast changing or unpredictable data models since it stores data in JSON format documents, which can have easily customizable and dynamic schemas. Using MongoDB

Mongoose:

Mongoose is an Object Data Modeling (ODM) library for MongoDB and NodeJS. Mongoose provides a straight-forward, schema-based solution to model our application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box. Mongoose is an Object Data Modeling (ODM) library for MongoDB and NodeJS. Mongoose provides a straight-forward, schema-based solution to model our application data. It includes built-in type casting, validation, query building, business logic hooks and more, out



of the box.

Fig 2:

Working of Mongoose in a server

Advantages of MERN Stack

Full-Stack JavaScript: Uses a single language for frontend, backend, and database operations.

Scalability: MongoDB's flexible schema and Node.js's non-blocking architecture enable scalable applications.

Fast Development: Reusable React components and Express.js's minimal setup accelerate development.

Rich Ecosystem: A large community and numerous libraries support rapid innovation and problem-solving.

The MERN stack is widely used for building modern web applications, including e-commerce platforms, social media sites, and enterprise solutions, offering a seamless development experience.

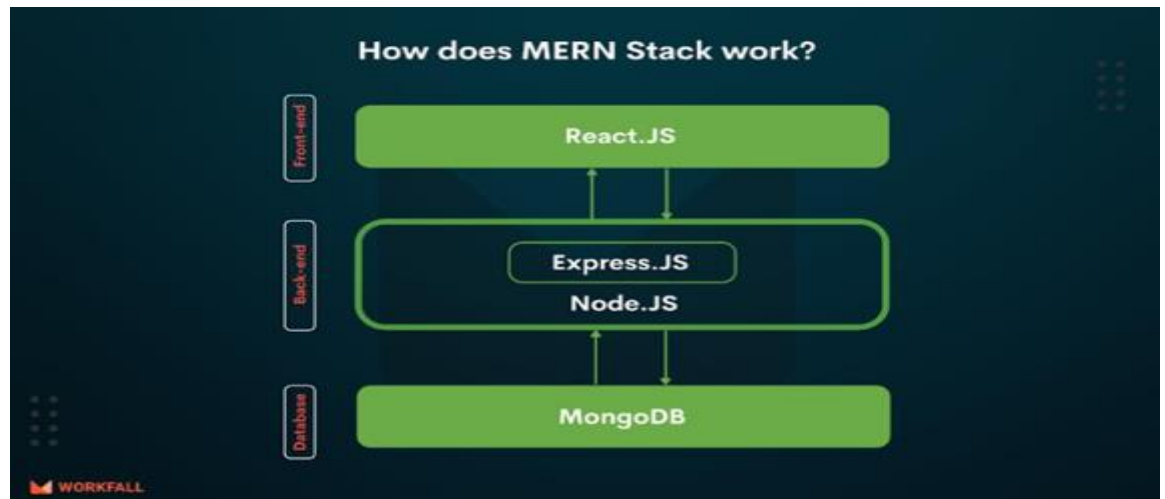


Fig 3: Diagram showing how the MERN stack works

Ethnus MERN stack Project Overview – Personal Finance Tracker

The **Personal Finance Tracker** is a full-stack web application developed as part of the Ethnus MERN Stack Internship Program. This project is designed to help individuals track their income, expenses, savings, and financial goals efficiently.

Developed using **MongoDB, Express.js, React.js, and Node.js (MERN stack)**, the system allows users to log transactions, categorize expenses, and generate insightful financial reports. It also includes **authentication and security** features to protect sensitive financial data.

The application was deployed on **Render**, utilizing **MongoDB Atlas** for scalable and secure cloud- based data storage. This ensures accessibility from anywhere while maintaining data integrity. By leveraging cloud hosting, the system is flexible, cost-effective, and capable of handling real-world financial management needs.

The project was built following **Agile methodologies**, ensuring iterative development, continuous improvement, and adherence to industry best practices. Through this project, I gained hands-on experience in full-stack web development, **API creation, authentication, financial data handling, and cloud deployment**, all of which contribute to my professional growth.

The **Personal Finance Tracker** is not just an academic project; it is a practical solution that helps users make informed financial decisions, reduce unnecessary expenses, and improve financial discipline.

Project Features

- **User Authentication:** Secure login/logout using OAuth, JWT, and session-based authentication.
- **Expense Tracking:** Log income and expenses with detailed descriptions.
- **Budget Management:** Set budgets and track spending habits.
- **Category Management:** Organize expenses into categories (e.g., food, rent, entertainment).
- **Savings and Goals:** Set financial goals and track savings progress.

- **Role-Based Access Control:** Different user roles for personal and shared financial management.
- **RESTful API Integration:** Backend services to handle financial data efficiently.
- **Responsive User Interface:** Built with React.js for seamless user experience.
- **Database Connectivity:** Uses MongoDB Atlas for secure and scalable storage.
- **Deployment on Render:** Ensures cloud accessibility and high availability.
- **Reports and Analytics:** Generate financial summaries with charts and insights.

PROJECT ARCHITECTURE

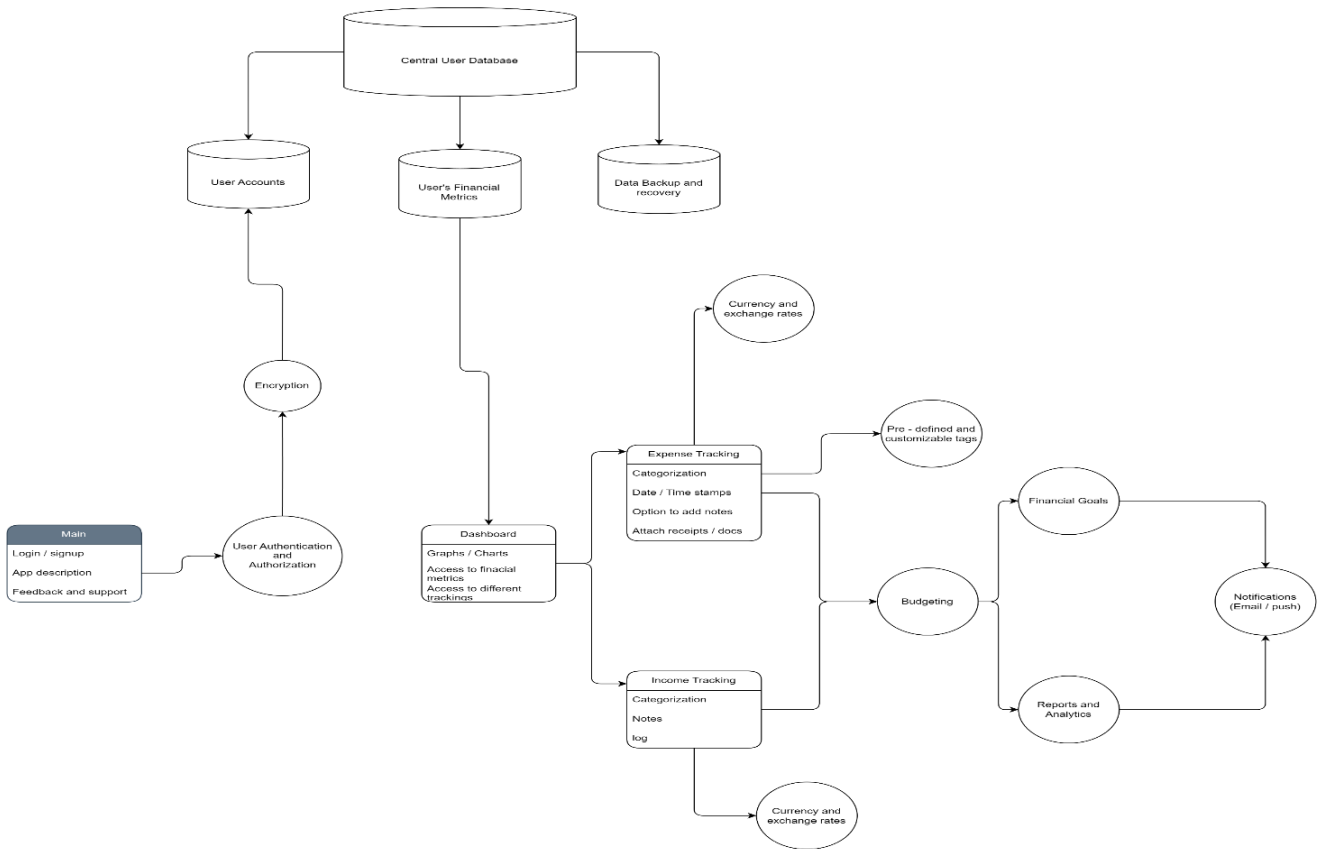


Fig 4: Process Flow Diagram of Personal Finance Tracker

The **Personal Finance Tracker** is built using the **MERN Stack (MongoDB, Express.js, React.js, and Node.js)** to provide a scalable, efficient, and modern web application. The architecture follows a structured approach to handling user data, financial transactions, and API integrations while ensuring security and performance.

System Components

The system follows a **three-tier architecture**, comprising the **frontend, backend, and database**.

1. Frontend (React.js)

- Developed using **React.js** for a dynamic and interactive user interface.
- Utilizes **React Hooks & Context API** for efficient state management.
- Implements **authentication mechanisms (JWT)** for secure access.
- **Expense Tracking & Budgeting:**
 - Users can add, update, and delete transactions.
 - Transactions are categorized for better financial insights.
- **Search & Filter Functionality:**
 - Implements a **real-time transaction search** using React state & debouncing.

- Provides filtering options by **category, date, and amount**.
- **Responsive UI design**, ensuring compatibility across devices.

2. Backend (Express.js & Node.js)

- Built with **Express.js**, a lightweight framework for handling **HTTP requests and API endpoints**.
- Implements **RESTful APIs** for **CRUD operations** (Create, Read, Update, Delete) on financial transactions.
- Handles **user authentication** using **JWT (JSON Web Tokens)** and OAuth.
- **Transaction API:**
 - Provides endpoints for users to **add, edit, delete, and retrieve transactions**.
 - Implements pagination for efficient transaction history viewing.
- Middleware for **data validation, error handling, and security**.

3. Database (MongoDB Atlas)

- **NoSQL database hosted on MongoDB Atlas** for high availability and scalability.
- Stores **user details, financial transactions, and budgeting data**.
- Uses **Mongoose ODM** for schema validation and data querying.
- **Indexing and Full-Text Search:**
 - Implements **MongoDB indexes** to optimize search queries.
 - Supports **partial matching for quick transaction lookup**.

Authentication & Security

- Implements **JWT-based authentication** for secure user access.
- Uses **bcrypt** for **password hashing** to enhance security.
- **Role-based access control (RBAC)** to differentiate between **admin and regular users** (if multi-user management is enabled).

Deployment & Cloud Integration

- The **frontend and backend** are deployed on **Render** for scalability and secure hosting.
- The **database** is managed on **MongoDB Atlas**, offering cloud-based data storage.

Workflow of the System:

1. User Authentication:

- Users sign in using **OAuth or JWT authentication**.
- Role-based access is assigned (**Admin/User** if applicable).

2. Transaction & Budget Management:

- Users can **log income and expenses** with detailed descriptions.
- Transactions are categorized for **better tracking (e.g., Rent, Groceries, Entertainment, etc.)**.
- Users can **set and track budgets** based on spending habits.
- **Search & Filter Functionality:**
 - Users can **search transactions** by name, category, or date.
 - Results **update in real time** with sorting and filtering options.

3. Financial Insights & Reports:

- Users can view **charts and graphs** for spending patterns.
- Generate **monthly/yearly expense reports** for financial analysis.

By integrating **robust financial tracking features**, the system empowers users to **manage their money effectively, track expenses, and achieve financial goals**.

OBJECTIVES ACHIEVED

The primary objectives of this internship were to gain hands-on experience in cloud computing and full-stack web development, improve problem-solving skills, and build a market-ready software product. These objectives were successfully achieved through structured training, project-based learning, and real-world application.

1. Ethnus MERN Full Stack Internship Program: This program offered comprehensive training in the MERN stack (MongoDB, Express.js, React, and Node.js), equipping me with a well-rounded skill set in both front-end and back-end development. I gained extensive exposure to designing responsive user interfaces, implementing secure authentication mechanisms (such as JWT or OAuth), developing RESTful APIs, and deploying applications to the cloud. The internship emphasized practical application, bridging theoretical knowledge with real-world execution, and honed my ability to navigate complex development workflows.

2. Personal Finance Tracker Project: This capstone project served as a proving ground for applying the skills acquired during the internship. By building a fully functional Personal Finance Tracker, I integrated features like real-time expense tracking, budget management, and robust user authentication. The project demanded proficiency in industry-standard technologies—such as MongoDB for database management, React Js for dynamic front-end development, and Node.js for server-side logic—while adhering to best practices like modular code design, version control with Git, and agile development methodologies. This hands-on experience reinforced my ability to tackle real-world challenges and deliver a polished, scalable solution. Some screenshots of the project are below.

Website Snippets:

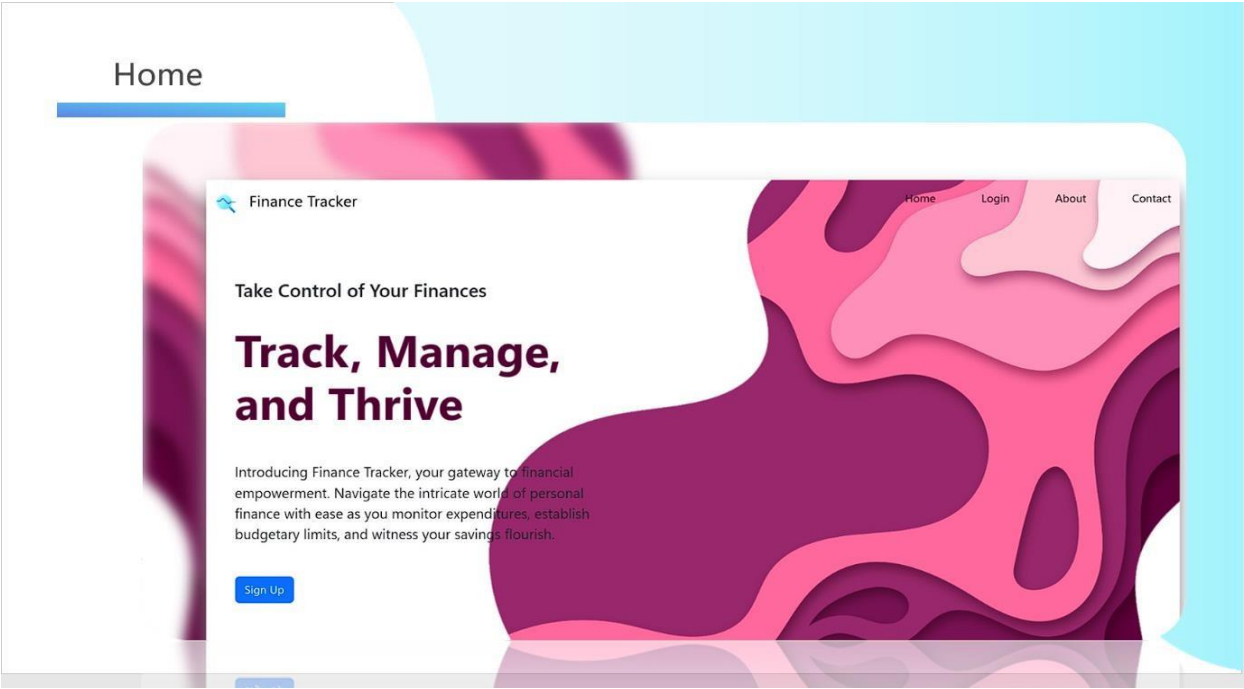


Fig 5: Home Page

Dashboard-Features

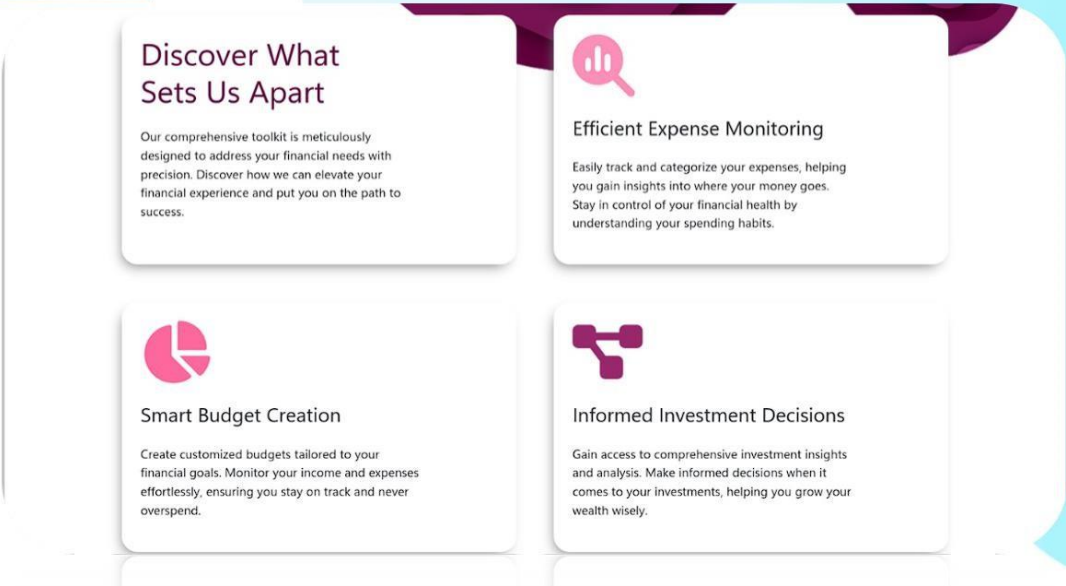


Fig 6: Dashboard Features

Main

Dashboard

Expense

Set Expense

Show Expense Log

Income

Set Income

Show Income Log

Budgeting

Set Financial Goals

Expense Log

Category: food

Date: 2000-08-18

Amount: 100000

Payment Method: bank_transfer

Currency: inr

Description: huhoo

Edit

Delete

Category: insurance

Date: 44444-04-04

Amount: 74747

Payment Method: none

Currency: inr

Description:

Edit

Delete

Fig 6.1: Expense Monitoring

Add Income

Category of Income:

Select category

Date of Income:

dd-mm-yyyy

Amount:

0

Currency Used:

Select currency

Payment Method Used:

☐

Cash Payment

☐

Bank Transfer

☐

Online Payment/UPI

☐

Cheque

☐

None

Description:

Fig 6.2: Income Operations

Set Budget

The form is titled 'Set Budget' and contains the following fields:

- Type of Budget:** A dropdown menu with 'Monthly' selected.
- Amount:** A text input field containing '10000'.
- Alert Percentage:** A text input field containing '80'.
- Description:** A text area containing 'Chips maangra hai'.

At the bottom right of the form is a pink 'Add' button.

Fig 6.3: Budget Setting

3. Deployment on Render and Netlify: Hosting the application on Render (for backend services) and Netlify (for the frontend) provided valuable insights into the intricacies of cloud- based deployment. This process involved configuring backend APIs, managing environment variables, and ensuring seamless integration with a MongoDB Atlas database. On the front end, I optimized static assets for performance and leveraged Netlify's continuous deployment features. Overcoming challenges like latency, CORS issues, and database connectivity enhanced my understanding of full-stack architecture, scalability, and the importance of monitoring application performance in a production environment.

SKILLS LEARNT

Technical Skills:

- **MERN Stack Development:**
 - **MongoDB:** Database design and query optimization.
 - **Express.js:** RESTful API development.
 - **React.js:** Front-end UI development and authentication.
 - **Node.js:** Server-side programming and middleware integration.
- **API Development:** Built and tested RESTful APIs for CRUD operations.
- **Authentication & Security:** Implemented OAuth, JWT authentication, and session-based security.
- **State Management:** Used Redux and React Context API for efficient state management.
- **Version Control:** Worked with GitHub for code collaboration and project management.
- **Cloud Deployment:** Hosted the web application on Render, integrating MongoDB Atlas as the database backend.

Professional & Soft Skills:

- **Project Management:** Worked in an Agile environment with sprint planning and daily stand-ups.
- **Problem Solving & Debugging:** Learned best practices for identifying and fixing bugs in complex applications.
- **Collaboration & Communication:** Engaged in peer code reviews and team discussions to improve code quality.

CHALLENGES FACED

1. Debugging Complex Code

While developing the Personal Finance Tracker, I faced issues with API responses, JWT authentication, and CORS errors in React-Node.js integration. These bugs were difficult to track as they often involved multiple layers of the stack.

How I Overcame It:

- Used Postman to test API endpoints systematically.
- Logged errors using console debugging and Morgan middleware in Express.js.
- Researched solutions on Stack Overflow, MDN, and AWS Developer Forums.

2. State Management in React

Managing global state in a large-scale MERN application was initially complex. Understanding Redux actions, reducers, and middleware took time. Issues like unnecessary re-renders and data inconsistency required an optimized approach.

How I Overcame It:

- Implemented Redux Toolkit to simplify state management.
- Used React DevTools to analyze state updates and performance bottlenecks.
- Explored Context API as an alternative for local state management.

3. Implementing Search Functionality in Inventory Management

One key feature of the Personal Finance Tracker was a search bar for filtering products based on name, category, or stock levels. Implementing this in both frontend and backend posed challenges, especially in optimizing queries for large datasets.

How I Overcame It:

- Used MongoDB aggregation and indexing to improve query performance.
- Implemented Debouncing in React to prevent unnecessary API calls.
- Enhanced UI/UX by adding loading indicators and error messages during searches.

4. Error Handling & User Feedback

A major challenge was ensuring proper error handling in both frontend and backend. Issues like invalid API responses, unhandled promise rejections, and user input validation needed a structured approach to avoid application crashes.

How I Overcame It:

- Used try-catch blocks in API calls to handle errors gracefully.
- Implemented a centralized error handler in Express.js to manage server-side errors.
- Displayed meaningful user-friendly error messages using React Toastify for better UX.

5. Deployment Issues

Deploying the full-stack application on Render, Netlify, and AWS introduced challenges, especially with database connectivity and environment variable management. The backend sometimes failed to connect with MongoDB Atlas due to IP whitelisting issues and incorrect CORS configurations.

How I Overcame It:

- Configured MongoDB Atlas Network Access to allow connections from Render.
- Used dotenv to manage environment variables securely.
- Deployed the frontend on Netlify and backend on Render, ensuring smooth integration.

RESULTS & OBSERVATIONS

The development of the Personal Finance Tracker yielded a robust and fully functional application, successfully incorporating essential features such as product tracking, order management, and user authentication. Beyond meeting its core functional requirements, the project showcased notable efficiency, scalability, and alignment with industry-standard development practices.

Feature Implementation & Functionality:

The system was designed to handle real-time inventory updates, track product availability, and process orders efficiently. User authentication and role-based access control were implemented to ensure secure management of inventory data.

• Industry Relevance & Commercial Viability:

The project followed industry best practices in software design, making it a potentially marketable product. Small businesses could adopt it to streamline inventory tracking, reduce manual errors, and enhance operational efficiency. The system's modular architecture allows for further enhancements and integrations, increasing its long-term value.

• Performance Optimization & Scalability:

By optimizing API performance and database queries, response times were significantly improved. Implementing indexing and query optimization techniques in MongoDB enhanced data retrieval speed. Load testing and caching strategies were explored to ensure the system could scale with increased user traffic.

• Deployment & Cloud Hosting Experience:

The project was deployed on multiple cloud platforms, including AWS, Render, and Netlify, providing hands-on experience with different hosting environments. Understanding deployment challenges such as server configurations, security policies, and resource allocation improved expertise in cloud-based application hosting.

• Team Collaboration & Learning Under Supervision:

Gained valuable experience working under supervision, following structured workflows, and adhering to project guidelines. Learned the importance of clear communication, taking feedback constructively, and implementing best practices while working in a team setting.

CONCLUSION

The Ethnus MERN Full Stack Internship Program provided a well-rounded experience in full-stack development and cloud computing, allowing me to gain hands-on expertise in building a real-world application. Through this internship, I successfully developed a Personal Finance Tracker, integrating essential features such as expense tracking, budget management, authentication, and financial insights. The project not only strengthened my technical skills but also helped me understand the challenges of deploying applications on cloud platforms like AWS, Render, and Netlify. This experience enhanced my proficiency in web development, API integration, and cloud-based application deployment.

Beyond technical knowledge, this internship emphasized the importance of debugging, optimizing API performance, managing state in React, and handling authentication securely. It also provided exposure to real-world development workflows, including code reviews and database optimization strategies.

The hands-on experience gained through this program has greatly enhanced my resume, making me a strong candidate for roles in web development and cloud computing. The Personal Finance Tracker is a scalable and potentially marketable product, showcasing the practical applications of MERN stack technologies in solving business problems.

This internship has laid a strong foundation for my future career in software development, equipping me with the necessary skills to tackle complex technical challenges. The knowledge gained will serve as a steppingstone for advanced certifications and further specialization in cloud computing and full-stack development.

BIBLIOGRAPHY

The following books and resources were referred to during the analysis and execution phase of the Ethnus project:

Books:

1. **"Full-Stack React Projects"** by Shama Hoque – Covers building scalable MERN stack applications.
2. **"Learning React: Modern Patterns for Developing React Apps"** by Alex Banks and Eve Porcello – Provides insights into React.js fundamentals and state management.

Online References:

- **MongoDB Atlas Documentation:** <https://www.mongodb.com/docs/atlas/>
- **React Official Documentation:** <https://react.dev/>
- **Node.js Documentation:** <https://nodejs.org/en/docs/>
- **Express.js Documentation:** <https://expressjs.com/>
- **Render Deployment Guide:** <https://render.com/docs>
- **Netlify Documentation:** <https://docs.netlify.com/>