



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

AI-Integrated MERN Stack Website for a Christian Learning NGO with Real-Time Chatbot Support

The domain of the Project
Full -Stack Development

Under the guidance of
Mr. Jasdeep Singh Hanspal
(Senior Technical Manager at Microsoft)

By

Mr. Sagar Raj (B.Tech IT Graduate)
Ms. Thejaswini (B.Tech CSE Graduate)
Mr. Karunesh Kumar Singh (B.Tech ECE Graduate)

Period of the project
December 2024 to April 2025



SURE TRUST

PUTTAPARTHI, ANDHRA PRADESH



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

DECLARATION

The project titled “**AI-Integrated MERN Stack Website for a Christian Learning NGO with Real-Time Chatbot Support**” has been mentored by **JASDEEP SINGH HANSPAL**, organized by SURE Trust, from December 2024 to April 2025, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. we declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

We Team members, hereby declare that we have solely worked on this project under the guidance of my mentor. This project has significantly enhanced my practical knowledge and skills in the domain.

Name

Mr. Sagar Raj- (**Backend Leader**)

Ms. Thejaswini- (**Frontend Leader**)

Mr. Karunesh Kumar Singh- (**Chatbot Leader**)

Mr. Aman Raj G25 FSD

Mentor

Mr. Jasdeep Singh Hanspal
Senior Technical Manager at Microsoft

Signature

Seal & Signature

Prof. Radhakumari
Executive Director & Founder



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)
SURE Trust



Table of Contents

1. DECLARATION	ii
2. TABLE OF CONTENTS	iii
3. EXECUTIVE SUMMARY	1
4. INTRODUCTION.....	2
4.1. Background and Context.....	2
4.2. Problem Statement	3
4.3. Scope.....	4
4.4. Limitations	4
4.5. Innovation	5
5. PROJECT OBJECTIVES	6
5.1. Project Objectives and Expected Outcomes	6
5.2. Deliverables.....	7
6. METHODOLOGY AND RESULTS	8
6.1. Methods/Technology Used	8
6.2. Tools/Software Used.....	9
6.3. Data Collection Approach.....	9
6.4. Project Architecture	10
6.5. Results	10
6.6. Final Project Hardware and Working Screenshots	11
6.7. GitHub Link	15
7. LEARNING AND REFLECTION.....	16
7.1. Learning and Reflection.....	11
7.2. Experience.....	13
8. CONCLUSION AND FUTURE SCOPE.....	20
8.1. Objectives.....	20
8.2. Achievements.....	20
8.3. Conclusion	20
8.4. Future Scope	22



Executive Summary

During the internship at SURE Trust, we worked on various frontend and backend development projects including mini-projects and one major MERN stack project for a Christian Learning NGO. These projects helped me understand real-world applications of web technologies including React.js, Node.js, MongoDB, and chatbot integration using Dialogflow. We gained practical experience in responsive design, API integration, state management in React, and building scalable and optimized backend systems.



Introduction

Background and Context

- The project is initiated by a Christian Learning NGO committed to offering high quality technical education in domains like MERN, MEAN, AWS, Azure, Android, Django, and Java.
- A web-based solution is needed to facilitate course visibility, student enrollment, and support services.
- The platform is being developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), chosen for its flexibility and performance.
- A key component of the system is the integration of an AI-powered chatbot to provide instant, intelligent assistance to users.
- The chatbot and website will operate seamlessly together, offering 24/7 guidance on course selection and batch schedules.
- GitHub is used for version control and collaboration among the development team.
- The system will prioritize responsive design and accessibility, aligning with WCAG standards to ensure usability for all.
- Cloud deployment on platforms like AWS or Azure will ensure availability and scalability.
- The project embodies both a technical challenge and a social mission to support digital literacy.
- By combining technology and education, the NGO aims to make a lasting impact on community-based learning.



Problem Statement

- There is currently no centralized, digital platform for learners to explore courses and receive timely support from the NGO.
- Manual processes for enrollment and course management are inefficient and lack scalability.
- Learners face confusion about available technologies, course schedules, and appropriate learning paths.
- The platform aims to offer an intuitive user experience through a structured navigation system.
- A key goal is to automate and enhance user support using an AI-powered chatbot.
- The system should dynamically fetch and display course information via API integration.
- Multi-platform support (web, mobile messengers) will broaden accessibility and engagement.
- The backend should provide secure data management and user authentication through JWT and OAuth.
- Administrative features such as a dashboard for managing courses and enrollments will streamline operations.
- Overall, the project strives to create a smart, responsive educational platform that enhances learning and simplifies administration.



Scope

- Frontend development using React.js and modern UI libraries (e.g., Tailwind CSS, Material UI).
- Backend services built with Node.js and Express.js, integrate MongoDB.
- RESTful APIs for data exchange between frontend, backend, and chatbot systems.
- AI chatbot development using frameworks such as Dialogflow, TensorFlow, or Rasa.
- User authentication using JWT and role-based access for admin functionalities.
- Cloud deployment on AWS, Azure, or similar platforms for high availability.

Limitations

- Chatbot responses may not fully handle complex or edge-case user queries without ongoing training.
- Voice command functionality is dependent on browser and device compatibility.
- Limited personalization features in the initial version (e.g., learning history or user recommendations).
- Cost constraints may limit full-scale deployment across all cloud services. o Real-time analytics or user behavior tracking is not included in the current project scope.



Innovation

- Development of an intelligent, voice-enabled chatbot integrated directly into the educational portal.
- Use of NLP (Natural Language Processing) and ML frameworks (e.g., TensorFlow, PyTorch) for smart conversation handling.
- Seamless multi-platform chatbot accessibility across major messaging apps.
- Real-time interaction using APIs to fetch enrollment and course data, enhancing responsiveness.
- Scalable cloud deployment to handle high user traffic and ensure consistent availability.
- Advanced DevOps practices including GitHub Actions for CI/CD automation and workflow optimization.
- Implementation of JWT and OAuth for secure and modern authentication practices.
- Adherence to accessibility standards (WCAG), ensuring the platform is inclusive and user-friendly.
- Strong focus on mobile-first design and cross-browser compatibility.
- Combines social mission with high-tech architecture, making it a unique educational technology solution.



Project Objectives

Project Objectives and Expected Outcomes

1. Clearly Defined Objectives and Goals of the Project

- Develop a user-centric and responsive educational website for a Christian Learning NGO using the MERN stack (MongoDB, Express.js, React.js, Node.js).
- Facilitate seamless course browsing, enrollment, and real-time information delivery through a dynamic front-end interface.
- Integrate a secure backend system with robust authentication (JWT, OAuth) and efficient NoSQL database management using MongoDB.
- Enable real-time data display and manipulation through RESTful API integration between frontend and backend services.
- Incorporate an AI-powered chatbot using NLP and ML models to provide 24/7 support for queries related to course selection, enrollment, and batch scheduling.
- Ensure accessibility compliance (WCAG standards) and design a cross-browser compatible, mobile-friendly interface.
- Deploy the application on scalable cloud platforms (AWS, Azure) for high availability and performance.
- Utilize GitHub for collaborative development, version control, and CI/CD automation using GitHub Actions.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

- Overall, aim to create a mission-driven digital solution that combines education, technology, and intelligent automation.

2. Expected Outcomes and Deliverables

- A fully responsive and visually appealing website designed using React.js, optimized for performance and accessibility.
- A secure backend developed using Node.js and Express.js, with a functional NoSQL database structure managed through MongoDB.
- End-to-end implementation of REST APIs for course content management, user enrollment, and dynamic data integration.
- A functional admin dashboard to manage batch schedules, course updates, and registered user data.
- An AI-integrated chatbot capable of answering FAQs, assisting in navigation, and handling voice/text queries related to course offerings.
- Multi-platform deployment of the chatbot, enhancing user reach and engagement across popular messaging apps.
- Deployment of the complete solution on a reliable cloud platform (AWS, Azure) ensuring continuous uptime and scalability.
- Source code and project documentation maintained in GitHub, following version control best practices and automated workflows.
- Comprehensive testing for both frontend and backend components to ensure reliability and performance under load.
- A modern, inclusive, and scalable educational platform that empowers users and supports the NGO's mission to expand access to technical education.



Methodology and Results

1. Methods / Technology Used

- The project employs the MERN Stack—MongoDB, Express.js, React.js, and Node.js— to create a full-stack, scalable web application.
- React.js is used for the frontend development to ensure a fast, responsive, and intuitive user interface.
- Node.js and Express.js power the backend, providing a secure and scalable server-side infrastructure.
- MongoDB serves as the NoSQL database to store course details, enrollment data, and user credentials efficiently.
- RESTful APIs facilitate smooth communication between frontend, backend, and the chatbot system.
- The platform uses JWT (JSON Web Token) and OAuth for user authentication and secure access control.
- For automation and version control, GitHub and GitHub Actions are utilized, ensuring collaborative and CI/CD-enabled development.
- Cloud deployment is supported via platforms like AWS or Azure for high availability and performance.
- The chatbot is built using Machine Learning frameworks like TensorFlow or spaCy, enabling intelligent user interactions.
- NLP (Natural Language Processing) powers chatbot conversations for real-time assistance with batch scheduling and FAQs.



2. Tools / Software Used

- **React.js:** Frontend framework used to build the UI with reusable components and efficient DOM rendering.
- **Node.js and Express.js:** Backend runtime and framework used to manage server-side logic and route handling.
- **MongoDB:** NoSQL database used for managing user enrollments, course data, and admin panel configurations.
- **Dialogflow:** AI tools used for chatbot development and Natural Language Processing.
- **Git & GitHub:** Used for source control, code collaboration, and version tracking across the development team.
- **GitHub Actions:** Automates testing, builds, and deployment pipelines in CI/CD workflows.
- **Cloud Platforms (AWS, Azure):** Used for hosting backend APIs, databases, and enabling scalable deployment.
- **Postman:** Used to test and debug REST APIs during development and integration.
- **Bootstrap / Tailwind CSS:** Frontend design libraries used to ensure a responsive and visually consistent user interface.
- **Docker (optional):** May be used for containerization if the project scales further in production environments.

3. Data Collection Approach (if applicable)

- The system collects user data during **course enrollment** via web forms built with React.js.
- Course-related data, batch schedules, and FAQs are stored in



MongoDB collections to be accessed dynamically.

- User credentials and access roles are captured securely using **JWT-based login forms** and saved in the backend database.
- The chatbot collects interaction data, such as common questions and user feedback, to train the ML model and improve NLP accuracy.
- Admins manually upload or update course details and batch schedules through the admin dashboard.
- Enrollment logs and user engagement metrics can be used to generate analytical reports (future implementation).
- API calls track real-time user queries and responses for support or troubleshooting.
- The chatbot's training set is derived from sample dialogues and frequently asked queries (FAQs) collected during beta testing.
- For privacy, user data is encrypted and access-controlled at the database level.
- All collected data is used strictly for enhancing user experience and optimizing chatbot responses.

4. Project Architecture

- The project follows a three-tier architecture: Frontend (React.js), Backend (Node.js/Express.js), and Database (MongoDB).
- The frontend communicates with the backend using REST APIs to fetch and display course details and manage user interactions.
- The backend handles all core business logic, including data processing, authentication (JWT/OAuth), and admin panel functionality.
- The MongoDB database stores user profiles, course catalogs, batch

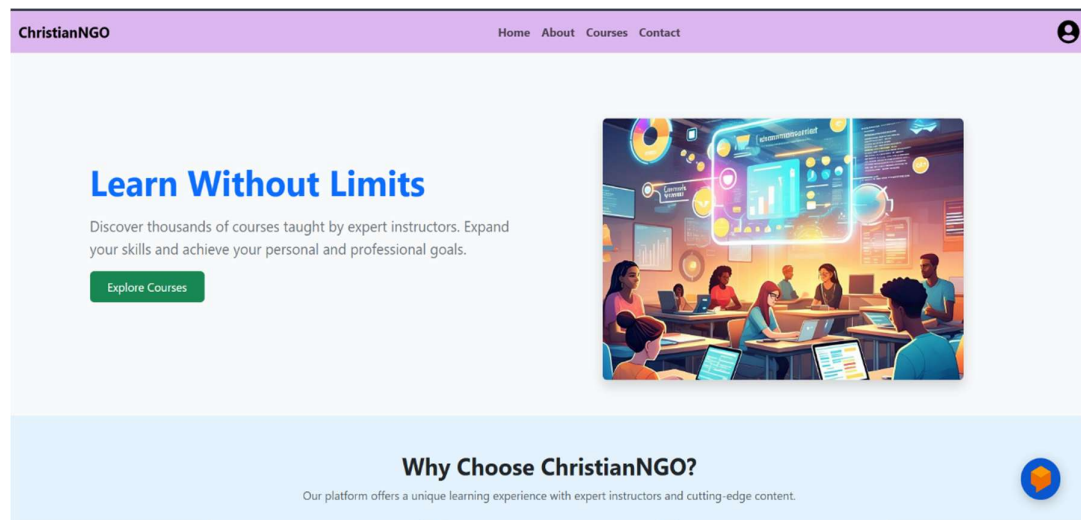


schedules, chatbot responses, and admin data.

- The AI-powered chatbot is integrated into the frontend and backend via API calls, enabling dynamic response generation.
- The chatbot uses ML/NLP engines to process user inputs and deliver accurate, conversational responses.
- GitHub is used for version control and code collaboration, while GitHub Actions manages CI/CD pipelines.
- The entire application is deployed on cloud platforms like AWS or Azure, ensuring availability and scalability.
- The architecture supports microservice principles, allowing modular growth and component reuse.
- The system ensures secure data flow through HTTPS, token-based access, and server- side encryption.

5. Final Project Working Screenshots with Explanation

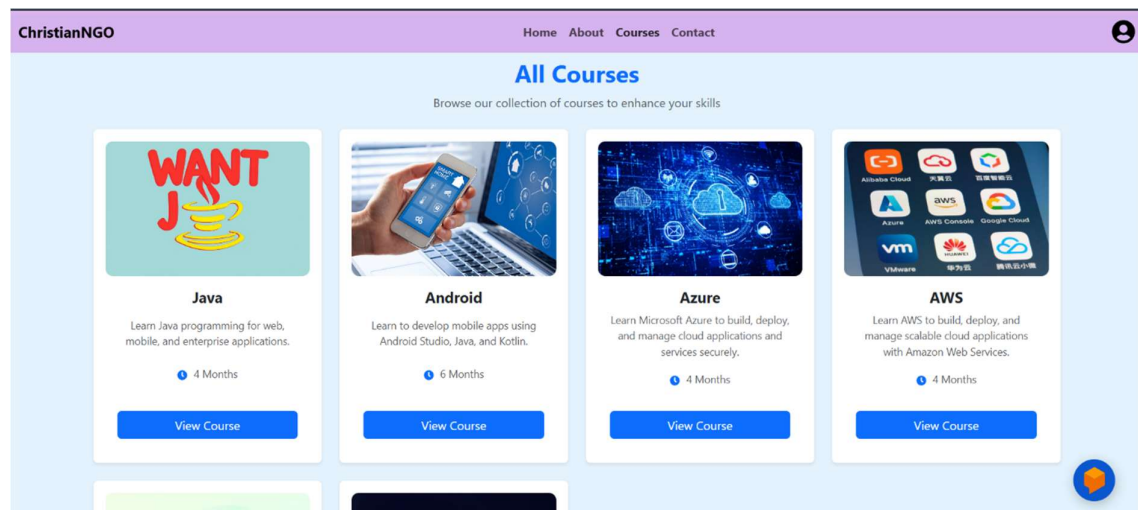
- **Homepage Screenshot:** Displays a clean and responsive landing page with featured courses and navigation options.



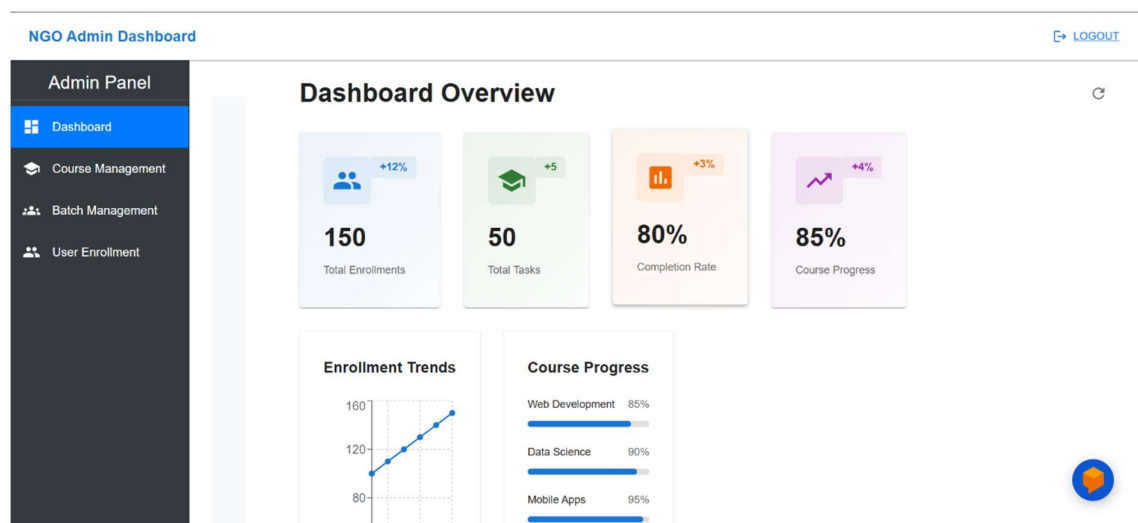


Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

- **Explanation:** Developed using React.js and styled with Tailwind CSS; optimized for speed and accessibility.
- **Course Detail Page:** Shows dynamic course listings with descriptions, prerequisites, and “Enroll Now” buttons.



- Explanation: Uses API integration to fetch course details from MongoDB.
- **Admin Panel Screenshot:** Allows admin users to manage course content, user enrollments, and batch schedules.

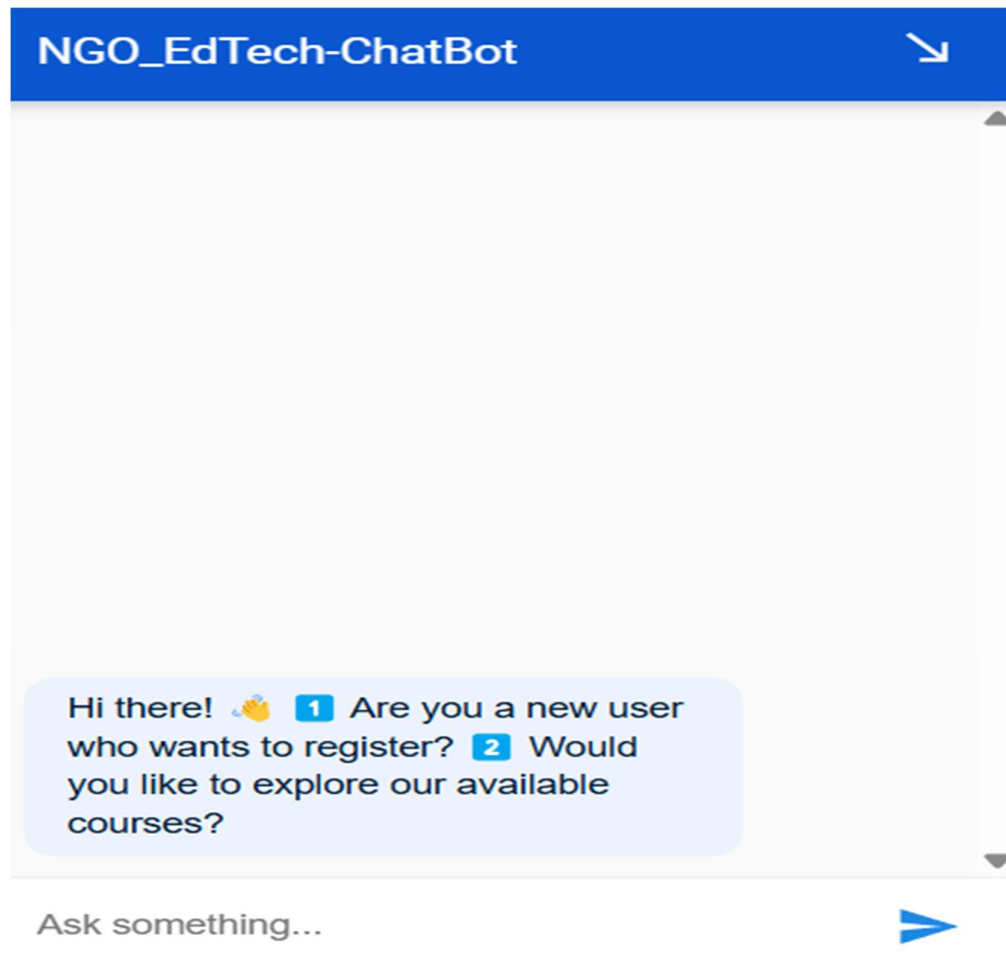


- Explanation: Secured with JWT-based login and accessible only to

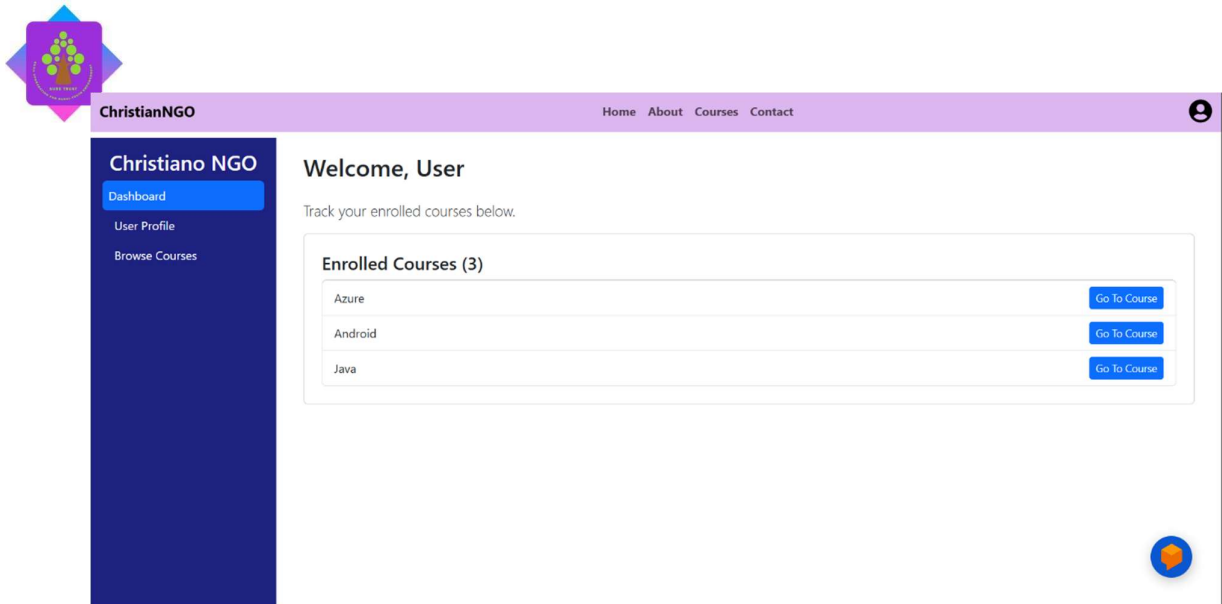


Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)
authorized users.

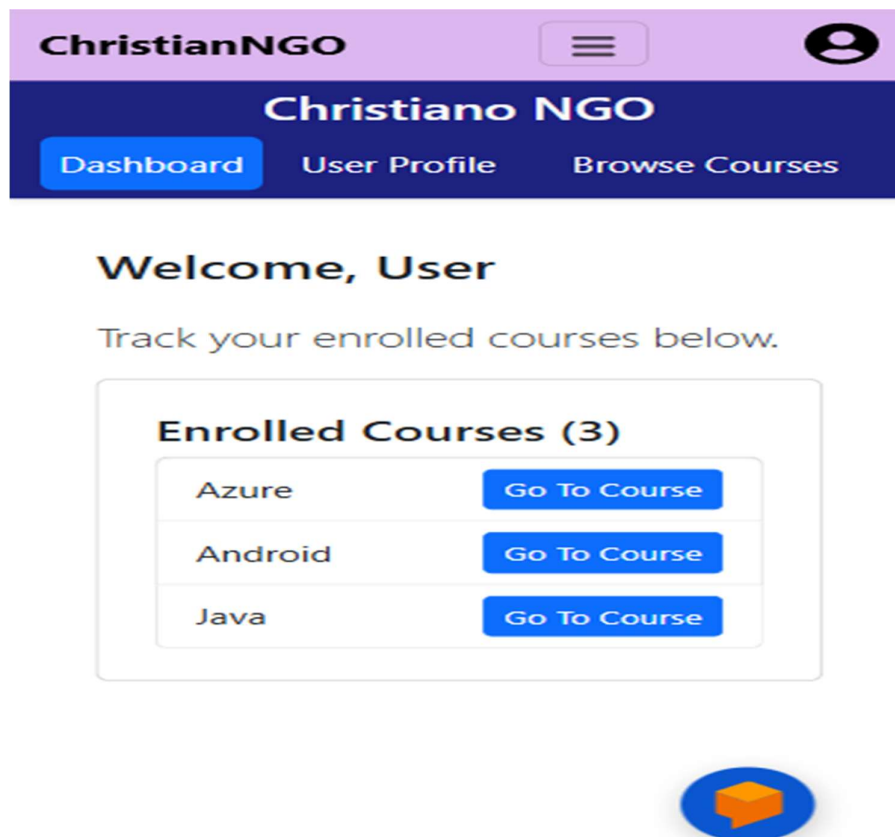
- **Chatbot Interface Screenshot:** Embedded chatbot providing assistance through voice/text inputs.



- Explanation: Integrated using Dialogflow or TensorFlow; supports real-time conversational support.
- **User Enrollment Page Screenshot:** Interactive form capturing student data with validations.



- Explanation: React form handling and API-based submission to the backend.
- **Mobile View Screenshot:** Demonstrates full responsiveness on mobile devices.





- Explanation: Ensures a smooth experience across platforms using responsive UI design.

6. Project GitHub Link

- GitHub Repository: https://github.com/Sagarraj2001/MERN_Education.git
 - Explanation: This repository contains the complete source code for the project, including frontend, backend, and chatbot components.
 - It is structured with proper folder hierarchy, documentation, and version history.
 - Includes README.md with setup instructions, features list, and contribution guidelines.
 - GitHub Actions is configured to manage automatic deployment and testing.
 - Future enhancements and issue tracking are also managed through the GitHub project board.



Learning and Reflection

1. Document Every Team Member's New Learnings (Technology, Management, etc.)

- **Frontend Developer:**

- Gained in-depth experience with React.js for building dynamic and responsive user interfaces.
- Learned how to apply UI/UX design principles to improve the usability of web applications.
- Mastered API integration to dynamically fetch course data from backend servers.
- Improved understanding of accessibility standards (WCAG) and cross browser compatibility practices.
- Learned how to secure frontend routes and implement JWT-based user authentication.
- Enhanced skills in working with GitHub, version control, and team-based collaborative workflows.
- Developed expertise in using Bootstrap and Tailwind CSS for responsive design.
- Understood the importance of performance optimization and SEO-friendly frontend code.
- Built skills in communicating and aligning design with backend data requirements.
- Adapted to a real-world agile workflow, managing timelines and feature prioritization.

- **Backend Developer:**

- Acquired hands-on experience in developing **secure, scalable APIs using Node.js and Express.js**.
- Learned to design and manage **NoSQL databases using MongoDB and Mongoose** effectively.
- Mastered **authentication and authorization** mechanisms using JWT and OAuth.



- Gained knowledge in **cloud deployment** using AWS and Azure platforms.
- Understood principles of **data encryption, access control**, and secure API endpoints.
- Learned to optimize backend performance for reliability and scalability.
- Practiced **CI/CD workflows** using GitHub Actions for deployment automation.
- Became proficient in using **testing frameworks like Jest** for backend validation.
- Gained real-time experience in handling data structures and building RESTful API services.
- Strengthened collaboration skills through integration with frontend and chatbot teams.

- **Chatbot Developer:**
 - Gained practical knowledge in **building AI-powered chatbots** using Dialogflow, TensorFlow, and PyTorch.
 - Understood core concepts of **Natural Language Processing (NLP)** for real-time query handling.
 - Learned to train, test, and optimize chatbot models for accuracy and contextual relevance.
 - Developed chatbot integration skills with **frontend UI and backend databases** using APIs.
 - Explored **voice-based interactions** and multi-platform deployments on Telegram, WhatsApp, etc.
 - Understood **chatbot behaviour tuning** based on user feedback and interaction data.
 - Implemented **GitHub workflows** for chatbot version control and updates.
 - Gained experience with **cloud deployment of chatbot services** for continuous availability.
 - Learned how to map user intents, FAQs, and conversational flows in a structured way.



- Strengthened teamwork by syncing chatbot features with core website development goals.

2. Document Every Team Member's Experience (Overall Experience)

• Frontend Developer:

- Found the experience enriching and challenging, especially while aligning design with functionality.
- Appreciated the opportunity to work with a mission-driven NGO and contribute to education.
- Learned how to balance **technical design and user experience**, a crucial skill in real-world projects.
- Experienced team collaboration using GitHub and learned how to manage frontend conflicts efficiently.
- Gained confidence in deploying responsive, accessible, and clean UI structures.
- Felt satisfied seeing real-time data integration and how the frontend brought the system to life.
- Took pride in contributing to a platform that helps learners across various domains.
- Encountered challenges with component structure and responsiveness but overcame them through research.
- Improved adaptability through feedback from backend and chatbot teams.
- Overall, described the project as a major milestone in frontend development career growth.

• Backend Developer:

- Experienced working on a **real-world scalable backend system** for the first time.
- Enjoyed building secure, production-ready APIs that drive a meaningful education platform.



- Gained new perspectives on how **backend systems directly influence frontend performance**.
- Faced challenges in integrating authentication and authorization securely but gained valuable experience.
- Learned to optimize queries and manage data schema for efficiency and security.
- Appreciated the structured development process with GitHub and automated deployment.
- Enjoyed team sync-ups and found cross-functional communication a key learning aspect.
- Developed a strong sense of responsibility, especially around **data integrity and system uptime**.
- Felt the experience improved both technical confidence and problem-solving mindset.
- Overall, described the backend development journey as transformative and highly fulfilling.
- **Chatbot Developer:**
 - Felt excited to contribute AI-powered solutions that directly help learners.
 - Enjoyed applying **ML and NLP knowledge** to real-world interactions through chatbot development.
 - Gained deeper insight into integrating APIs, training datasets, and deploying scalable bots.
 - Found multi-platform deployment (Telegram, Messenger) challenging but rewarding.
 - Appreciated how conversational design impacts user satisfaction and learning guidance.
 - Learned to refine bot responses through testing and user feedback cycles.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

- Found teamwork invaluable in aligning chatbot data with the admin panel and website flow.
- Developed creative problem-solving skills when handling vague or multi-intent queries.
- Improved knowledge in AI model tuning and version control practices.
- Overall, described the experience as eye-opening and a gateway into practical AI integration.



Conclusion and Future Scope

1. Recap of Objectives and Achievements

- The main objective was to design and develop a comprehensive educational platform for a Christian Learning NGO using the **MERN stack**.
- The team successfully built a fully functional and responsive website with a clean UI developed using **React.js** and styled with modern CSS frameworks.
- Key features such as course listings, enrollment forms, and batch schedule integration were implemented using **dynamic RESTful APIs**.
- The backend, powered by **Node.js** and **Express.js**, was integrated with **MongoDB**, offering secure and scalable data management.
- The system supports **JWT- and OAuth-based authentication**, ensuring secure access for users and administrators.
- A robust **admin panel** was created to manage users, course content, and scheduling effectively.
- A cutting-edge **AI chatbot** was integrated using frameworks like TensorFlow/Dialogflow, enhancing user support through NLP.
- The chatbot was made accessible across platforms including the website, **WhatsApp, Telegram, and Facebook Messenger**.
- Version control and CI/CD pipelines were implemented using **GitHub** and **GitHub Actions** to streamline development and deployment.
- The project achieved its goal of delivering an end-to-end education management solution that is both scalable and user-friendly, supporting the NGO's mission.

2. Future Scope of the Project

- **Personalized Learning Paths** can be introduced using AI algorithms to recommend courses based on user history and preferences.
- Integration of **video conferencing tools** (like Zoom or Google Meet) can enable live classes and interactive sessions within the platform.



- **Mobile application development** can expand accessibility and engagement, especially for remote learners using smartphones.
- The chatbot can be enhanced with **multi-language support**, improving inclusivity for users from different linguistic backgrounds.
- Real-time **analytics dashboards** for admins can be added to monitor learner performance, engagement, and course popularity.
- Implementing **payment gateway integration** would enable paid courses or donations directly through the platform.
- **Gamification features** such as badges, quizzes, and leaderboards could boost learner motivation and retention.
- The current system could be modularized further to adopt a **microservices architecture**, increasing maintainability and scalability.
- **AI-based auto-suggestions** for FAQs and chatbot responses can further improve user satisfaction and efficiency.
- Future collaborations can bring in **certification systems** or partnerships with other educational bodies for expanded learning recognition.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)