Sachin Suresh

Bangalore sachin.samprit@gmail.com +919886083024 Sachin Suresh Sachin-0001 Portfolio

Education

Dayananda Sagar College of Engineering *Oct 2023-Present*
BE in Computer Science

- **CGPA:** 9.4/10.0

Achievements

BotCraft *Nov-2024*
Runners

- Developed a Discord bot using Python to foster community engagement for remote teams, securing a runner-up position in the 'Bot Craft' competition. Implemented features that enhanced user interaction and automated routine tasks.

The Great Bengaluru Hackathon *Mar-2025*
Shortlisted

- Engineered 'FairFare,' a full-stack application using the MERN stack (MongoDB, Express, ReactJS, NextJS) to address ride cancellations. Trained and deployed machine learning models in Python to predict cancellation probability, aiming to reduce rider inconvenience during peak hours

Technologies

Languages: C++, Java, JavaScript, Python

Frameworks/Libraries: ReactJS, NextJS, Express, TailwindCSS, Scikit-learn, TensorFlow, Keras, Streamlit

Tools: VS Code, Git, GitHub, DagsHub, MLFlow

Databases: MongoDB, MySQL

Projects

Code Chronicles *visit repo*

- CodeChronicles could be positioned similarly as a smart coding knowledge aggregation platform. It can curate, categorize, and present trending coding resources, tutorials, and discussions in a structured, user-friendly format. This way, it helps users stay informed with personalized learning paths and an intuitive coding experience.
- Tools Used: NextJS, JavaScript, TailwindCSS, MongoDB

FairFare *visit repo*

- FairFare aims to reduce ride booking cancellation during peak hours which add to the inconvience of the public. The Ml models we have trianed and deployed work efficinetly in communion to solve the isssue.
- Tools Used: NextJS, JavaScript, TailwindCSS, MongoDB, Python

Chat with Docs *visit repo*

- **C**hat With Docs is an intelligent document assistant platform that enables users to upload and interact with documents using natural language queries. Leveraging advanced language models and retrieval-augmented generation (RAG), it allows seamless extraction of insights from large documents like PDFs or manuals. Users can ask questions, search contextually, and receive accurate, conversational responses — improving productivity and document comprehension.
- Tools Used: LangChain, Python, Streamlit, OpenAI API, FAISS, PyMuPDF