

IMRAN.I

Arakkonam, Tamil Nadu | iimraniimran4@gmail.com | +91 7397677968 | iimran-portfolio.netlify.app
linkedin.com/in/i-imran09 | github.com/i-imran09

Summary

Front-End Developer

-I'm a front-end developer who enjoys creating responsive and user-friendly websites using React.js, JavaScript, HTML, CSS, and Bootstrap.

-I hold a Bachelor's degree in Computer Science, which has helped build a strong foundation in programming and web development.

-I enjoy learning new things like responsive UI design and API integration, and I pick them up quickly while giving my best effort to grow my skills.

Skills

- **Frontend:** HTML, CSS, JavaScript, React.js, Bootstrap, API Integration, State Management
- **Version Control:** Git, GitHub
- **Deployment:** Netlify
- **Others:** Prompt Engineering

Education

Thiruvalluvar University, BSC in Computer Science

Sep 2022 – Apr 2025

- **College:** Arakkonam Arts and Science College
- GPA: 7.1/10

Tamilnadu StateBoard, Bio-Maths

Jun 2018 – Mar 2020

- **School:** St Andrew's Hr Sec School Arakkonam
- Percentage: 43.17

Projects

New-Hunt

news-hunt-latest.netlify.app

- **Tech Stack:** React.js, Bootstrap, GNews API
- **Description:** News-Hunt is a responsive web application that displays the latest Indian news articles. Users can browse news by categories such as Business, Entertainment, General, etc. Each article includes a title, summary, and a link to the full source.
- **What I Did:** Connected GNews API to fetch live news data. Used Bootstrap CDN for styling and responsive design. Created 3 main components: NavBar, NewsBoard, and NewsItem. Used useState and useEffect to fetch and manage news articles. Passed data using props and displayed it using map() method. Handled category and language filters through state and props.

Reac-Tube – YouTube Clone

reac-tube.netlify.app

- **Tech Stack:** React.js, CSS, JavaScript, React Router, YouTube V3 API
- **Description:** Reac-Tube is my first cloning project, built as a YouTube replica. It displays videos from YouTube using the YouTube V3 API, allowing users to watch them and see the publisher's name, likes, and comments.
- **What I Did:** Created two pages: Home and Video, and set up routing in App.jsx. Built a Nav component with three sections (left, center, right) and designed it with a dynamic hamburger menu. Used state to toggle the sidebar visibility when the hamburger menu is clicked. Displayed videos in the Feed component by fetching data from the YouTube V3 API. Added Link tags in Feed to route to /video/category.id for video pages. On the Video page, created Play Video and Recommended Video components and fetched API data to display them. Added a Link to the Nav logo for routing back to the home page.