Rahul Dey

6290323403 | deyr279@gmail.com | PortFolio | LinkedIn | GitHub | Credentials

Education

Woolf University — California
M.Sc in Computer Science — *March 2024 – Present*

Alma Better — Bengaluru
Full Stack Web Development — *July 2023 – March 2024*

NSHM Knowledge Campus Kolkata — Kolkata, West Bengal
B.Sc in Hospitality And Hotel Administration — *Sept 2020 – June 2023*

Experience

Web Development Trainee (MERN stack) — March 2023 – March 2024
Alma Better — *Bengaluru*

- Skills: MERN, Search Engine Optimization, DSA, Databases, Monitoring, Team work, Business Communication.
- Gained hands-on experience in front-end with Back-end programming languages and tools such as React.js, Next.js, Node.js and using databases such as MongoDB and MariaDB.
- Created and optimized web applications, utilizing front-end frameworks like React and back-end frameworks like Express .
- Learned about best practices, including user authentication, federated login/register and data encryption, to develop and deploy high-quality and secure web applications.
- Secured the 2nd place among the top 3 students in the cohort of 232+ students , and worked as an SME.

Industrial Trainee — Sept 2022 – Feb 2023
The Lalit Great Eastern Kolkata — *Kolkata, West Bengal*

Projects

Entertainment App — Live Demo | GitHub

- ∗ Tech-Stack Used: React, JavaScript, HTML, CSS, Tailwind CSS, MongoDB, Node.js, Express.js, TMDB API
- ∗ User-Centric Design: The Entertainment App boasts a user-centric design, featuring engaging pages tailored to enhance user experience. The dynamic Home page serves as a central hub, showcasing recent and trending movies and TV series in an organized grid system, captivating users from the moment they log in. Dedicated pages for Movies and TV Series offer a diverse selection of content across various genres, catering to different preferences and ensuring users can easily find what they're looking for. Additionally, the Bookmarks page provides a personalized space for users to curate their own entertainment library, allowing them to bookmark their favorite movies and TV series for easy access later on. With these user-friendly pages, the app is designed to keep users engaged and satisfied with their entertainment choices.
- ∗ Secure Authentication and Data Protection: With user authentication via Sign Up and Login pages, coupled with the implementation of secure authentication and data encryption for protecting user information, the project ensures a high level of security. Best practices such as input validation and the use of environment variables for sensitive data protection further bolster the app's security measures.
- ∗ Scalable Architecture and Collaborative Development: The project follows best practices in its architecture, employing a modular code structure for maintainability, code commenting for readability, and version control with Git for efficient team collaboration. Continuous integration and deployment processes streamline testing and deployment, ensuring a smooth development lifecycle and facilitating scalability as the app grows.

Portfolio Website — Live Demo | GitHub

- ∗ Tech-Stack Used: React, JavaScript, HTML, CSS, Bootstrap, MongoDB, Node.js, Express.js, Nodemailer
- ∗ Skills Showcase: Highlighted my expertise in full-stack web development, including proficiency in languages such as HTML, CSS, JavaScript, and frameworks like React.js and Node.js.
- ∗ Project Gallery: Presented a curated selection of my completed projects, showcasing diverse functionalities, design aesthetics, and problem-solving abilities.

- About Me Section: Provided insight into my background, education, professional experience, and passion for web development, fostering a personal connection with visitors.
- Contact Form and Downloadable Resume: Implemented a user-friendly contact form for inquiries and collaboration opportunities, along with a downloadable resume for easy access to my qualifications and accomplishments.

Flashcard Generator Live Demo | GitHub

- Tech-Stack Used: React, JavaScript, HTML, CSS, Tailwind CSS, Redux, Formik, Yup, React-router-dom, React-icons
- Seamless Flashcard Creation: Utilizing React, JavaScript, HTML, and CSS, our project offers users a user-friendly platform to effortlessly generate personalized flashcards.
- Intuitive Interface: With Formik and Yup, users can easily fill out a form interface to create flashcards with titles, descriptions, and multiple terms. Dynamic addition and removal of terms enhance user flexibility.
- Structured Overview: Employing Redux and React-router-dom, the "My Flashcards" section provides a structured overview of all created flashcards, enabling easy navigation and management.
- Enhanced Sharing and Accessibility: Integration of React-icons and Tailwind CSS enables sharing functionality via unique links and a download feature to convert flashcards into PDF files. Offline study is supported with the option to print physical copies, ensuring accessibility and convenience, even without an internet connection.

To-Do-App Live Demo | GitHub

- Tech-Stack Used: React, JavaScript, HTML, CSS, React-Dom, UUID, Jest-Dom
- Developed using React, JavaScript, HTML, and CSS with React-Dom for rendering components. Utilized UUID for unique task identification and Jest-Dom for testing components, ensuring reliability and scalability of the ToDo app.
- Project Description: Created a ToDo app allowing users to effortlessly manage daily tasks. Users can seamlessly add and remove tasks, enabling them to stay organized and productive throughout their day.
- The live demonstration of the app with github repository is given.

Weather-App Live Demo | GitHub

- Tech Stack Used: React, JavaScript, HTML, CSS, React-Dom, Jest-Dom, Weather API
- Utilized React, JavaScript, HTML, and CSS for frontend development, along with React-Dom for rendering components and Jest-Dom for testing. Integrated Weather API to fetch real-time forecasts, current Celsius temperature, and wind speed data for any location.
- Description: Introduces a user-friendly weather companion, offering instant access to accurate weather updates in a single interface. Simplifies the process of staying informed about the weather conditions, making it convenient for users to plan their activities accordingly.

Get-Youtube-Subscribers Live Demo | GitHub

- Tech-Stack Used: React, JavaScript, HTML, CSS, NodeJS, Express.js, Mongoose, MongoDB, Chai, Mocha, Dotenv, Mochawesone
- YouTube Subscriber Tracker: This project utilizes React, JavaScript, HTML, and CSS to create a user-friendly interface for fetching and displaying detailed subscriber information from YouTube channels.
- Node.js Backend: The backend is built with Node.js, Express.js, Mongoose, and MongoDB. It handles requests from the frontend, fetching subscriber data from YouTube and storing it in a MongoDB database.
- API Endpoints: The project provides several API endpoints:
  GET /subscribers: Returns an array of subscribers as objects, providing detailed subscriber information.
  GET /subscribers/names: Returns an array of subscribers with only two fields: name and subscribedChannel.
  GET /subscribers/:id: Returns a subscriber object with the given id. If the id does not match, it responds with a status code 400 and an error message.
- Testing and Validation: The project includes testing using Chai and Mocha, ensuring that the API endpoints function correctly and handle errors appropriately. Additionally, it utilizes Dotenv for environment configuration and Mochawesome for generating detailed test reports in Node.js.

Amazon-Clone Live Demo | GitHub

- Tech-Stack Used: React, HTML, CSS

- Utilized React for dynamic rendering and interactivity, enhancing user experience with features like product listings, search functionality, and responsive design.
- Employed HTML and CSS to meticulously craft the layout and design elements, ensuring visual fidelity to the original Amazon interface while honing skills in front-end development.

Technical Skills

Languages: JavaScript, HTML, CSS, Data Structures and Algorithms, SQL(Maria DB), No SQL(MongoDB)
Frameworks: React, jQuery, Tailwind CSS, Bootstrap, NextJS, Express.js, Node.js, REST API
Developer Tools: Git, Github, AWS, VS Code, Visual Studio
Libraries: Mongoose, React Router, Express Validator, Axios, Redux, Jest, Chai, Mocha