

PUSKAR ROY

+91 7449585365

puskarroy600@gmail.com

Barasat - 700121 ,North 24 PGS

puskarroyuitbu.netlify.app



SUMMARY

As a first-year Computer Science Engineering student at UIT, the University of Burdwan, I possess a strong foundation in programming languages and web development technologies. I have hands-on experience with React.js, Node.js, MongoDB, Express.js, Python, Java, JavaScript, and C.

EDUCATION

UIT ,The University Of Burdwan

B.E. in Computer Science & Engineering
2022-2026

SKILLS

- Python
- Java
- C
- Java Script
- HTML
- CSS
- React Js
- Node Js
- Express Js
- Mongo DB
- Google Firebase

ACHIEVEMENTS

- Rank 10 - In **Smart Bengal Hackathon** RCCIIT.

LANGUAGES

- English
- Hindi
- Bengali

PROJECTS AND EXPERIENCE

Pathway

A Real Time Chat Web Application

I developed a beginner-level real-time chat application using the MERN (MongoDB, Express.js, React.js, Node.js) stack with Socket.IO. Leveraging my skills in front-end and back-end development, I built a user interface with React.js that enables users to register, log in, and engage in real-time chat conversations. The application utilizes Socket.IO to establish WebSocket connections for instant message delivery. By integrating MongoDB for data storage and Express.js for server-side handling.

IntelliSched

A Time Table scheduling Gui application.

To Solve This Problem We Use CSP(Constraint Satisfaction Problem) To Generate A Schedule, Then To Satisfy Those Hard And Soft Constraints We Use A Concept From Evolutionary Computing That Is Called Genetic Algorithm. From Using Genetic Algorithm We Find The Best Schedule. For Making GUI We Use A python Library Called Custom Tkinter And In Backend We Use Google Firebase (Firebase Authentication And Cloud Firestore)

Authentication API

ts A Simple Rest API using Express Js & Node Js

Developed a secure authentication API using Express.js, MongoDB, bcrypt.js for password hashing, and JWT tokens for login. Implemented user registration and login endpoints with proper error handling. Integrated bcrypt.js for secure password storage by hashing the passwords before storing them in the database. Implemented JWT token-based authentication for user login, allowing users to authenticate and access protected routes. Utilized MongoDB as the database to store user information securely.