

Subedar Aarthi

Full Stack | subedaraarthi@gmail.com | +91 9393273000

LINKEDIN : www.linkedin.com/in/subedar-aarthi | GITHUB : <https://github.com/AarthiSubedar/>

SKILLS

- PROGRAMMING LANGUAGES : Java
- FRONT-END : HTML,CSS,Javascript,Reactjs
- BACKEND DEVELOPMENT : Springboot ,Servlet
- DATABASE : Oracle,Postgre SQL
- API DEVELOPMENT : REST API
- TOOLS : Git,Postman

EDUCATION

BSC Computer Science

TARA GOVERNMENT DEGREE & PG COLLEGE,
SANGAREDDY

Aug 2023

CGPA : 9.32

Intermediate,MPC

NRI JUNIOR COLLEGE

May 2020 | Hyderabad

Percentage : 89.5 %

CBSE

JAWAHAR NAVODAYA VIDYALAYA

May 2018 | Wargal

Percentage : 78.6%

CERTIFICATES

- Udemy – HTML and CSS for Beginners from Basic to Advance.
- Udemy – HTML and CSS for Beginners from Basic to Advance.
- Sathya Technologies - Java Full Stack

EXPERIENCE

INTERNSHIP: FULL STACK WEB

DEVELOPMENT | 1 Month | BHARAT INTERN

LANGUAGES KNOWN

- TELUGU
- HINDI
- ENGLISH

SUMMARY

Aspiring web developer eager for entry-level opportunities. Trained in Java full stack development, ready to contribute and grow. Enthusiastic about joining dynamic teams and making an impact.

PROJECTS

BANKING MANAGEMENT SYSTEM

Developed and implemented a Banking Management System, focusing on CRUD operations. Streamlined customer interactions, enabling efficient account creation, deposits, and secure fund transfers.

- Used Spring MVC to handle incoming requests and manage the flow of data between the front-end and back-end.
- Used HTML and CSS for structuring and styling your web pages respectively.
- Utilized JSP for dynamic content generation and integrating Java code with HTML for rendering dynamic data.
- Deployed Spring Boot application along with the JSP frontend on a server. used containers like Apache Tomcat or embedded containers provided by Spring Boot for deployment.
- Used database (MySQL) to store banking-related data such as customer information, account details, and transaction history.

PRODUCT MANAGEMENT SYSTEM

- Utilized Spring Boot for the back-end to handle data processing, business logic, and database interactions. ReactJS will serve as the front-end framework for building dynamic user interfaces.
- Designed clear and concise RESTful APIs for communication between the front-end and back-end. Follow REST principles, such as using HTTP methods appropriately (GET, POST, PUT, DELETE) and utilizing meaningful URI endpoints.
- Implemented secure authentication and authorization mechanisms to protect user data.
- Set up a robust data management system using relational databases (Oracle) for storing product information and user data.