Vaibhav Naik

Full-Stack-Developer

vaibhavnaik32275@gmail.com

01vaibhavnaik

01vaibhavnaik-Portfolio

01vaibhavnaik-Git

8329694138

Goa

EDUCATION

Computer Science And Engineering
SDIT , Mangalore
8.2 CGPA | 2025

SKILLS

- **Core Java**
- **CSS**
- **HTML**
- **JavaScript**
- **BootStrap**
- **React**
- **MySql**
- **PostgreSQL**
- **Servlets & JSP**
- **JDBC**
- **Lombok**
- **Spring (Spring-MVC , JPA)**

TOOLS

- **Git**
- **Intellij IDEA**
- **VSCode**
- **Mysqlworkbench**
- **Eclipse**
- **PostgreSQL**

LANGUAGES

- **English**
- **Konkani**
- **Hindi**

CAREER OBJECTIVE

Motivated Web Application Developer with training in **Java Full Stack technologies**, including **Spring (Spring MVC, JPA)**, **HTML, CSS, JavaScript, Bootstrap**, and **MySQL**. Previously worked on backend-driven web applications using Spring-based architectures. **Currently working on frontend development using React and database operations with PostgreSQL** for **Goa Online Service–based applications**, gaining hands-on experience in real-time development environments. Eager to contribute technical expertise, collaborate with cross-functional teams, and deliver reliable and scalable software solutions.

WORK EXPERIENCE

Software Intern
X-workz, Bangalore

Gained hands-on experience in Core Java, web technologies, and SQL, with practical exposure to tools like Git, IntelliJ IDEA, and VS Code. Maintained coding discipline by uploading projects to GitHub using Git Bash and IntelliJ. Contributed to the development of a software application using Spring, strengthening expertise in Java Full Stack development.

Software Intern
Accolade Tech Solution, Mangalore

Completed training in Java fundamentals with a focus on object-oriented programming concepts, including inheritance, encapsulation, abstraction, and polymorphism, building a strong foundation for web application development.

TECHNICAL SUMMARY

1. CORE-JAVA

- Strong in **Object-Oriented Programming (OOPs):** Encapsulation, Inheritance, Polymorphism, Abstraction.
- Practical experience in **Interfaces & Collections Framework** – List (ArrayList, LinkedList), Set (HashSet, LinkedHashSet, TreeSet), Map, Queue.
- Proficient in **Java 8+ features** – Lambda Expressions, Stream API (functional-style operations), and Optional class for null-safe handling.
- Solid understanding of **Exception Handling** (try, catch, throw, throws, finally) and **Multithreading**.
- Hands-on with **JDBC** for DB connections and CRUD operations.
- Experience with **Servlets & JSP** using MVC architecture (DTO, Service, Repository layers).
- Knowledge of **Spring Framework** (IoC, dependency injection, bean lifecycle, component wiring).
- Familiar with **JPA (Java Persistence API)** for ORM mapping and CRUD using annotations.
- Comfortable using **Lombok** annotations like **@Data, @Getter, @Setter** to reduce boilerplate code.

2. FRONTEND TECHNOLOGIES

- **HTML, CSS, JavaScript:** Strong understanding of semantic HTML, different CSS types (inline, internal, external), and responsive layout design.
- Implemented **form validations** in JavaScript with input checks and dynamic user feedback.
- **Projects:** Built a Web-Page using **Bootstrap** HTML And CSS for UI structuring and styling.

- **Bootstrap:** Hands-on with grid system, navbar, buttons, and cards. Built a **responsive website** ensuring mobile-friendly and consistent design.
- **React** Basic to intermediate experience in React, working with **functional components**, **props**, and **state management using useState**. Familiar with handling **side effects using useEffect**, component lifecycle behavior, and building **reusable UI components**. Implemented dynamic rendering and user interactions in React-based projects.

3. BACKEND TECHNOLOGIES

- **Servlets & JSP:** Experienced with request-response lifecycle, fetching form data using request parameters, generating response pages, redirection, and layered architecture (Service & Repository).
- **JDBC:** Practical experience in connecting Java applications with **MySQL**, saving form data to tables, and retrieving records via SQL queries.
- **Spring (Spring MVC, JPA):** Built applications using dependency injection, component wiring, ORM mapping, and CRUD with annotations.
- Implemented **form validation in Spring** using **@Valid annotation** with JSR-303/JSR-380 bean validation (Hibernate Validator).
    - **@NotNull, @NotEmpty, @NotBlank** – for mandatory fields
    - **@Size** – for length restrictions
    - **@Email** – for validating email formats
    - **@Min, @Max** – for numeric range checks

4. DATABASE

(1) My-SQL

- Proficient in creating tables and executing **CRUD operations** (Create, Read, Update, Delete).
- Experienced in **integrating MySQL with Java applications** using JDBC.

(2) PostgreSQL

- Proficient in creating tables and executing **CRUD** operations (Create, Read, Update, Delete) using **PostgreSQL**.
- Experienced in integrating PostgreSQL with React applications through REST APIs.
- Familiar with consuming backend APIs in React to fetch, display, and update data stored in PostgreSQL.
- Basic understanding of data flow between React frontend and PostgreSQL via backend services.

PROJECTS

1. Personal Portfolio Website

- Developed a fully responsive **portfolio website** using **Bootstrap** to ensure modern, mobile-first layouts.
- Implemented **Anime.js** for smooth animations and interactive UI elements, enhancing user experience.
- Designed clean and structured layouts to showcase personal details, skills, and projects in a professional format.
- Integrated **Emailable.js API** to enable visitors to send emails directly through the portfolio, allowing me to easily reach and connect with them.

2. SignUp-SignIn Page (Spring MVC + JPA)

- Designed and implemented a **web application** using **Spring, Spring MVC, JPA, and Bootstrap**.

- Enabled seamless **data flow from frontend to backend**, storing records in the database with proper dependency management.
- Configured entities using **JPA annotations** (**@Entity, @Table, @Id, @GeneratedValue, @Column**) for ORM mapping and structured database persistence.
- Applied **backend validation** using **@Valid** annotations and **frontend validation** with JavaScript to ensure data accuracy and prevent invalid inputs.
- Implemented **secure password handling** by integrating **Spring Security** and using BCryptPasswordEncoder for encryption, ensuring robust authentication and data protection.
- Integrated **JavaMail** with **Mkyoung implementation** to send confirmation emails to users upon successful signup.
- Followed **MVC architecture** with layered design (**DTO, Service, Repository**) to ensure clean separation of concerns and maintainable code.
- Implemented **custom queries** to perform **update** and **delete** operations on database records efficiently.

3. **Hospital Management Web Application (Spring MVC + JPA)**

- Designed and implemented a hospital management web application using **Spring, Spring MVC, JPA (Hibernate), and Bootstrap**.
- Developed role-based modules including Admin and Doctor, enabling secure admin login with **OTP-based authentication**.
- Implemented admin functionality to register doctors, edit doctor profiles, and manage doctor availability slots.
- Enabled **seamless data flow between frontend and backend**, persisting application data in the database with proper dependency management.
- Configured **ORM mappings** using JPA annotations **(@Entity, @Table, @Id, @GeneratedValue, @Column)** to ensure structured and efficient database persistence.
- Applied **backend validation** using **@Valid** annotations and frontend validation using JavaScript to ensure data accuracy and prevent invalid inputs.
- Implemented secure password encryption using Spring Security with **BCryptPasswordEncoder** to protect user credentials.
- Integrated **JavaMail (Mkyong implementation) to send OTP** and confirmation emails during admin login and doctor registration.
- Followed MVC architecture with layered design **(Controller, DTO, Service, Repository)** to maintain clean separation of concerns.
- Implemented custom JPA queries for efficient update and delete operations on doctor records and scheduling data.