**Personal Portfolio Website**

**Project Report**

Project Title: Personal Portfolio Website  
Name: Sandeep Kumar  
Degree: B.Tech (Electronics Engineering)  
Institution: SLIET Longowal  
Year: 2025  
Contact: sink10704@gmail.com | +91 7007935226  
GitHub: https://github.com/krSandip

# Abstract

This project presents a modern and responsive Personal Portfolio Website developed to showcase academic achievements, technical skills, and completed projects. Built using Spring Boot and Thymeleaf for the backend and server-rendered views, and standard front-end technologies (HTML, CSS, JavaScript) for layout and interactivity, the website aims to provide a professional online presence for a student/developer.

# Introduction

An effective online portfolio is essential for students and professionals to present their work, demonstrate skills, and make a positive impression on recruiters and collaborators. This portfolio is designed to be clean, fast, and accessible, focusing on clarity of information and aesthetic design. The site is structured using Thymeleaf fragments for reusability and maintainability.

# Motivation & Problem Statement

Many developers rely on static resumes or scattered project links. A centralized portfolio consolidates projects, education, achievements, and contact information into a single, shareable URL. The project addresses the need for a deployable, easy-to-update portfolio that can be hosted with minimal setup.

# Objectives

* Design a responsive, accessible portfolio website.
* Showcase projects, skills, education and achievements in a professional layout.
* Demonstrate practical knowledge of Spring Boot and Thymeleaf.
* Provide contact mechanisms and links to external profiles (GitHub, LinkedIn).
* Keep the code modular using fragments and reusable components for easy maintenance.

# System Overview

The website follows a simple server-rendered model: the Spring Boot backend handles incoming HTTP requests and maps routes via controllers. Thymeleaf templates receive model data and render HTML views that are served to clients. Static resources (CSS, JS, images) are served from the application's resource folders. The application is lightweight and can be deployed to cloud hosts like Render or Heroku.

# Modules / Components

1. Home - Landing section with greeting, hero visual, and primary navigation.
2. About - Background, skills summary, and image showcasing the developer's persona.
3. Projects - Cards highlighting important projects with tags, links to code and live demos.
4. Experience / Education - Timeline displays academic history and key milestones.
5. Achievements - Cards summarizing recognitions and project milestones.
6. Contact - Contact methods and a hosted contact form to receive messages.

|  |  |
| --- | --- |
| **Icon** | **Technology & Description** |
|  | **Spring Boot** Java-based framework used to build production-ready backend services. Handles routing, dependency injection, and application lifecycle. |
|  | **Thymeleaf** Server-side Java template engine used to generate HTML views with reusable fragments and expressions. |
|  | **HTML5** Markup language used to structure web pages and content. |
|  | **CSS3** Styling language used for layout, responsive design, and visual theming. Custom CSS provides the modern look and animations. |
|  | **JavaScript** Client-side scripting for interactions, smooth scrolling, observers, and minor dynamic behavior. |
|  | **Git** Version control system used to track changes and manage the project repository. |

1. Footer & Navigation - Persistent footer with quick links and social icons; responsive navigation for mobile.

# Technologies Used

# Implementation Details

Project structure follows a standard Spring Boot layout:  
- src/main/java/com/sandip/portfolioproject -> Java application and controller classes  
- src/main/resources/templates -> Thymeleaf templates and fragments (home.html, fragments/\*)  
- src/main/resources/static -> CSS, JS, and image assets  
  
Key files:  
- PortfolioprojectApplication.java: Application entry point.  
- HomeController.java: Maps routes for home and related pages.  
- templates/home.html: Main template that includes fragments for modular sections.  
  
The UI uses a modular fragments approach for reusability; individual sections are maintained in separate fragment files, which simplifies updates and ensures consistent layout across pages.

# Screenshots

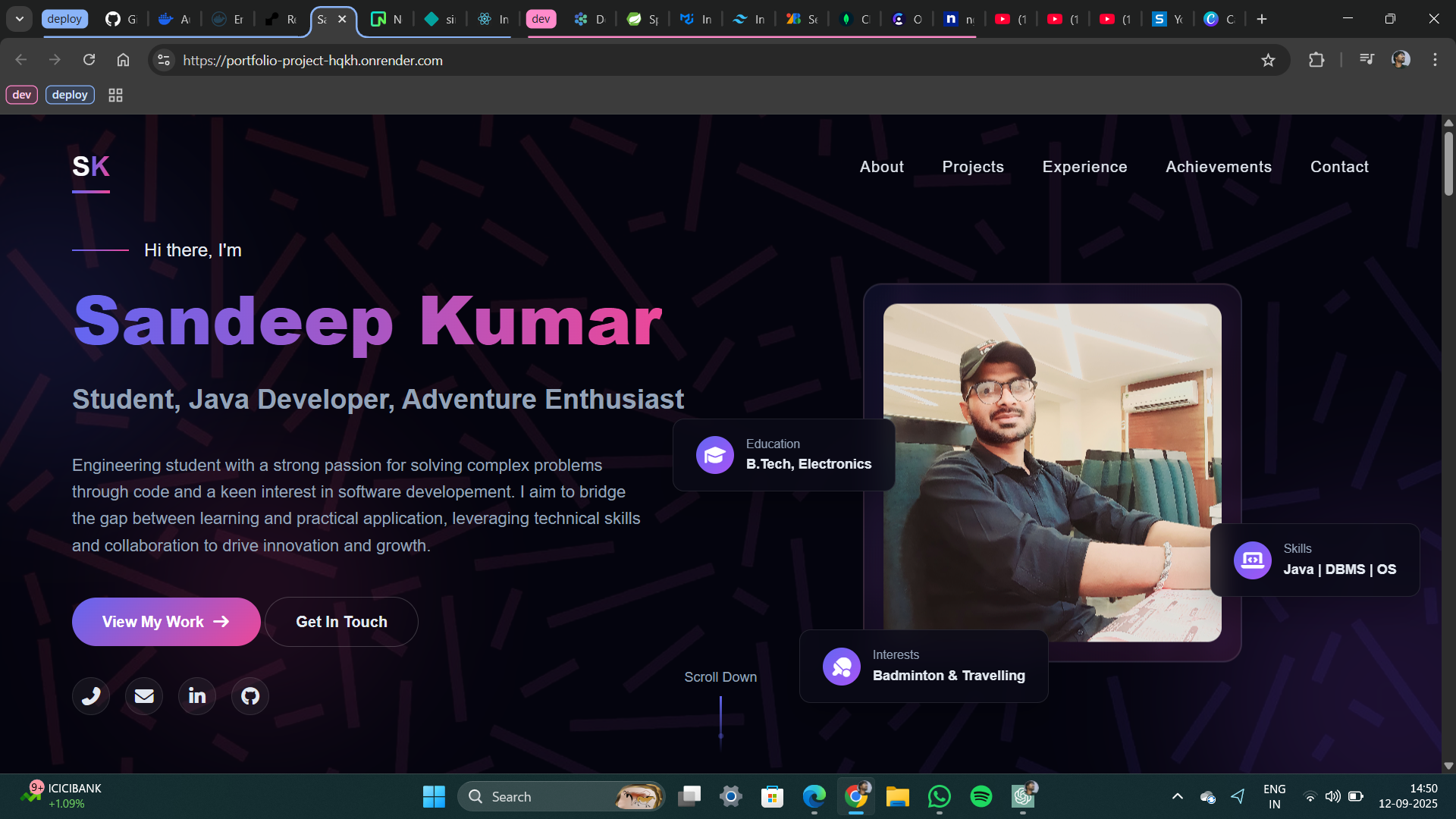


Figure 1: Website screenshot (section view).

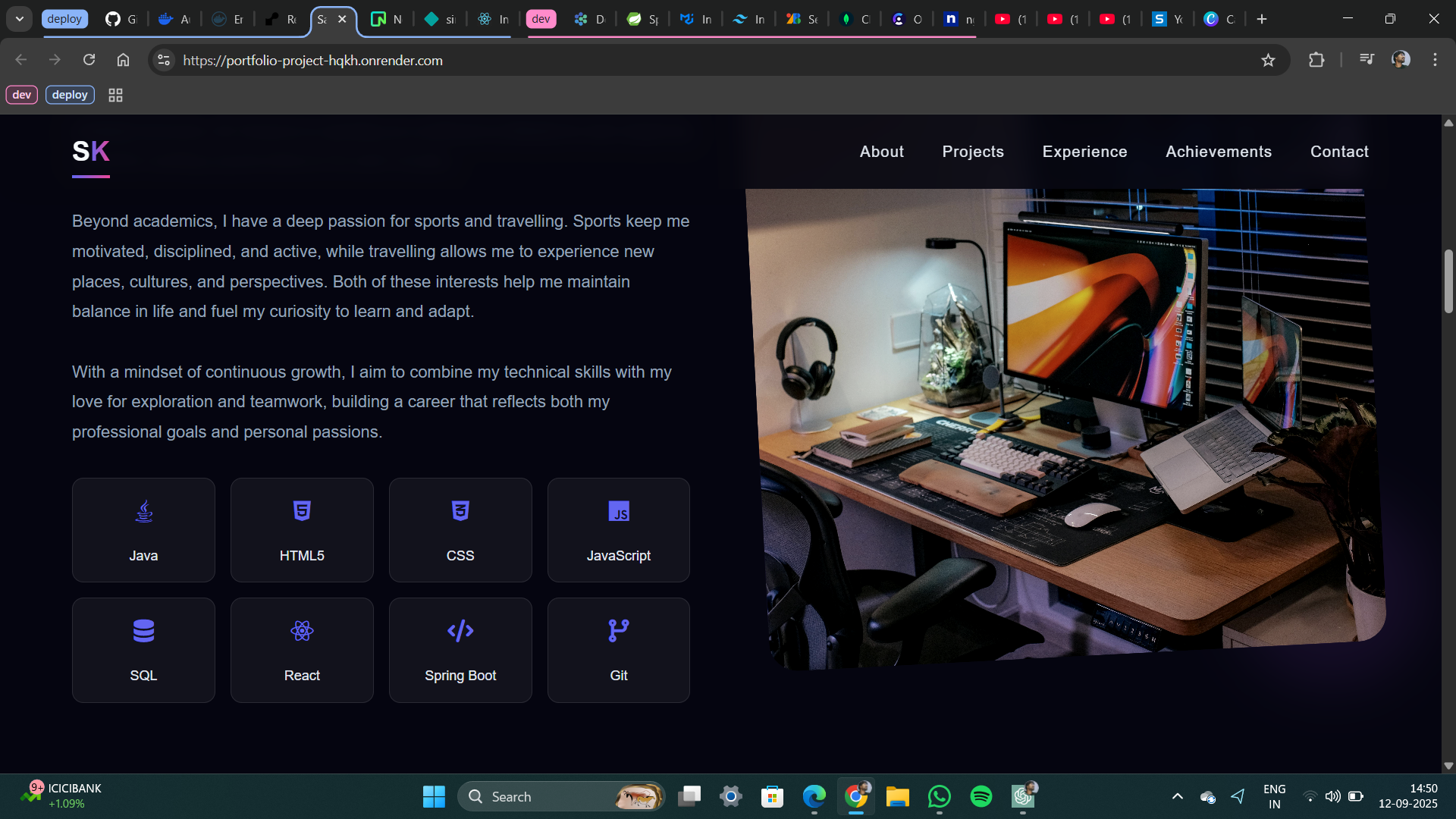


Figure 2: Website screenshot (section view).

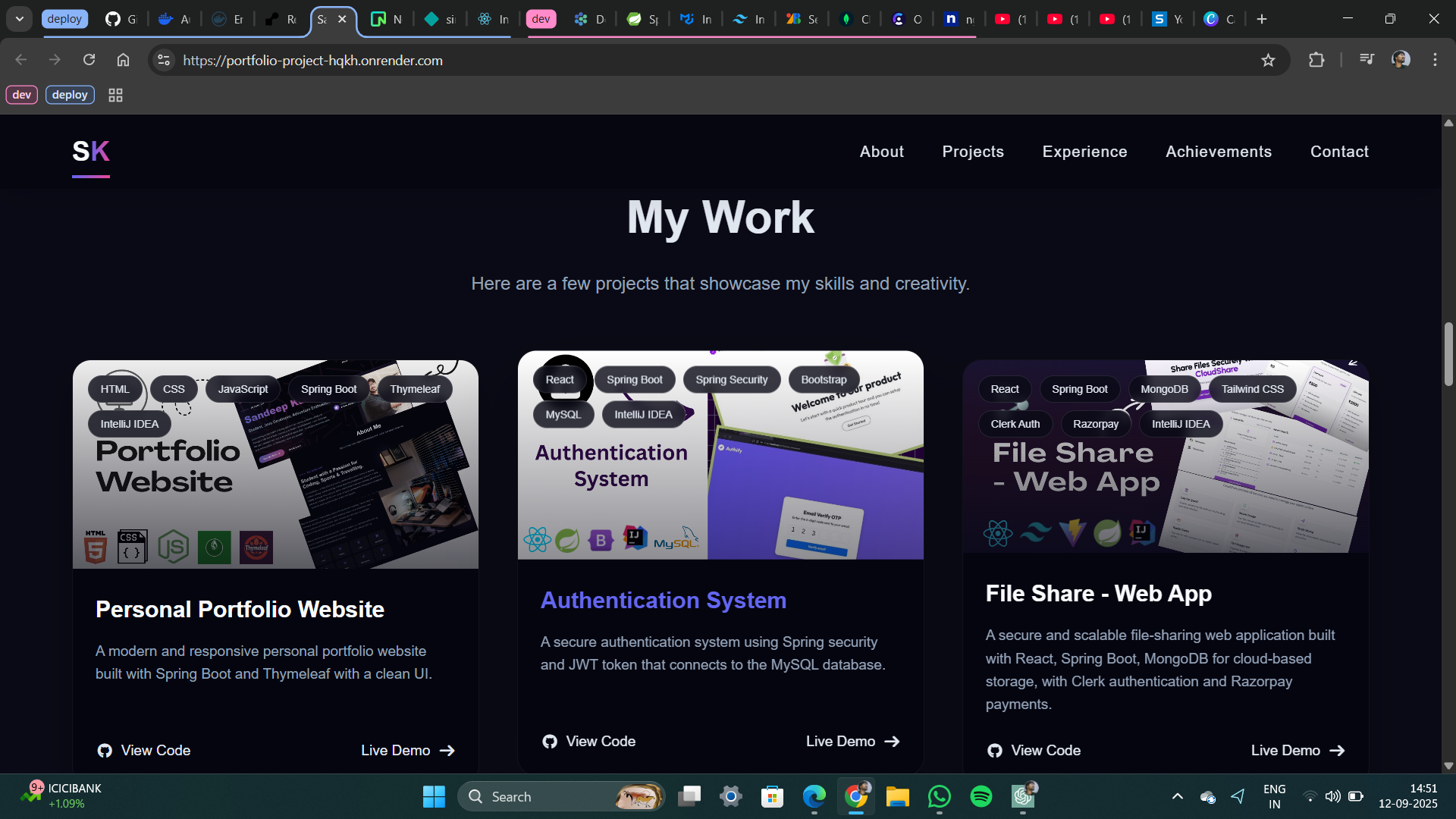


Figure 3: Website screenshot (section view).

# Testing & Deployment

The application can be run locally using Java 17 and Maven. Typical commands:  
1) mvn clean package  
2) mvn spring-boot:run  
  
For deployment, lightweight cloud services (Render, Heroku) can host the JAR or a Docker container. Ensure environment variables and resource routing (static assets) are configured correctly on the hosting platform.

# Conclusion

This Personal Portfolio Website project demonstrates the ability to design and implement a professional, responsive web presence using Spring Boot and Thymeleaf along with modern front-end practices. The modular structure and clean styling make the site easy to maintain and extend in future iterations.

# Future Enhancements

* Add an admin dashboard to update projects and content dynamically (database-backed).
* Implement a blog or articles section with markdown support.
* Add analytics and visitor tracking dashboards.
* Introduce server-side rendering optimizations and SEO improvements.
* Implement unit/integration tests and CI/CD pipeline for automated deployment.

# References

1. Spring Boot Documentation (https://spring.io/projects/spring-boot)

2. Thymeleaf Documentation (https://www.thymeleaf.org/)

3. MDN Web Docs for HTML/CSS/JavaScript (https://developer.mozilla.org/)