# Developer Documentation for Sporting Goods Webpage

## Table of Contents

1. Introduction

2. Technologies Used

3. Setup and Installation

4. Folder Structure

5. Coding Guidelines

6. Version Control with GitHub

7. Collaboration Tools

8. Database Design

9. Testing

10. Deployment

11. Contact Information

---

## 1. Introduction

This developer documentation provides guidelines and instructions for developing and maintaining the Sporting Goods webpage. The webpage is designed to facilitate the online sale of sporting goods, featuring a user-friendly interface and secure payment processing.

## 2. Technologies Used

- Frontend: HTML, CSS, JavaScript

- Backend: Java Servlets, JSP (JavaServer Pages)

- IDEs: VS Code, IntelliJ, NetBeans

- Version Control: GitHub

- Collaboration: Discord, Teams, Email

- Database: MySQL

- Web Server: Apache Tomcat

## 3. Setup and Installation

1. Clone the GitHub repository to your local machine.

2. Set up MySQL database and import provided schema.

3. Configure your IDE (VS Code, IntelliJ, or NetBeans) for Java development.

4. Install Apache Tomcat for serving JSP and Servlets.

5. Configure your local development environment with appropriate environment variables.

6. Run the application locally for testing.

## 4. Folder Structure

```

sporting-goods-webpage/

│

├── src/ # Java source files

│ ├── com/

│ │ ├── example/

│ │ │ ├── servlets/ # Servlet classes

│ │ │ ├── utils/ # Utility classes

│

├── web/ # Web content

│ ├── WEB-INF/

│ │ ├── jsp/ # JSP files

│ │ ├── lib/ # External libraries

│ │ ├── web.xml # Web deployment descriptor

│

└── README.md # Project documentation

```

## 5. Coding Guidelines

- Follow consistent naming conventions.

- Write clean and readable code.

- Use meaningful comments for better understanding.

- Ensure proper error handling and validation.

- Keep code modular and reusable.

## 6. Version Control with GitHub

- Create feature branches for each new feature or bug fix.

- Commit changes with descriptive messages.

- Pull latest changes frequently.

- Push changes to remote branches regularly.

- Follow the Git Flow or similar branching strategies.

## 7. Collaboration Tools

- \*\*Discord\*\*: Real-time communication for quick discussions and issue resolution.

- \*\*Teams\*\*: Collaboration hub for team discussions, meetings, and file sharing.

- \*\*Email\*\*: For formal communication, documentation sharing, and announcements.

## 8. Database Design

- Utilize MySQL for data storage.

- Follow normalization principles for efficient database design.

- Document the database schema for reference.

## 9. Testing

- Write unit tests for backend logic.

- Perform integration tests for end-to-end functionality.

- Test cross-browser compatibility and responsiveness.

- Use debugging tools provided by IDEs for troubleshooting.

## 10. Deployment

- Deploy the application to a staging environment for final testing.

- Once tested, deploy to the production server.

- Use continuous integration and deployment (CI/CD) pipelines for automation.

- Monitor application performance and logs post-deployment.

## 11. Contact Information

For any queries or assistance, please contact:

- \*\*Project Lead\*\*: [Name] - [Email]

- \*\*Technical Support\*\*: [Name] - [Email]

---

This developer documentation provides a comprehensive guide for developing, deploying, and maintaining the Sporting Goods webpage. Ensure to adhere to the outlined guidelines and best practices throughout the development lifecycle.