# Technical Documentation: Sporting Goods Webpage

## Introduction

This technical documentation outlines the development and management process for a sporting goods webpage. The project utilizes various tools and technologies including GitHub for version control, web browsers for testing, Discord and Microsoft Teams for communication, email for notifications, VS Code and IntelliJ NetBeans for code development, and JSP (JavaServer Pages) and Servlets for dynamic web content. Additionally, the project involves database design for storing information related to sporting goods.

## Tools and Technologies Used

1. \*\*GitHub:\*\* Version control system for managing codebase and collaboration.

2. \*\*Web Browsers:\*\* Used for testing and viewing the webpage.

3. \*\*Discord and Microsoft Teams:\*\* Communication platforms for team collaboration and coordination.

4. \*\*Email:\*\* Used for notifications and communication.

5. \*\*Visual Studio Code (VS Code):\*\* Code editor for front-end and back-end development.

6. \*\*IntelliJ IDEA and NetBeans:\*\* Integrated development environments for Java development.

7. \*\*JSP and Servlets:\*\* Technologies for creating dynamic web content in Java.

8. \*\*Database Design:\*\* Designing and implementing a database to store information related to sporting goods.

9. \*\*Tomcat Server:\*\* A Java Servlet container used for deploying and running Java Servlets and JSP.

## Development Workflow

### 1. Setting up the Development Environment

- Install required software: VS Code, IntelliJ IDEA, Tomcat Server.

- Clone the project repository from GitHub to your local machine.

### 2. Front-end Development

- Use VS Code to edit HTML, CSS, and JavaScript files for front-end development.

- Test the webpage in various web browsers to ensure compatibility.

### 3. Back-end Development

- Use IntelliJ IDEA or NetBeans for Java development.

- Implement Servlets and JSP pages for dynamic content generation.

- Connect to the database to retrieve and store information about sporting goods.

### 4. Database Design

- Design the database schema to store information such as product details, categories, and user data.

- Use appropriate tools (e.g., MySQL Workbench) for database design and management.

### 5. Version Control with GitHub

- Commit and push code changes to the GitHub repository.

- Collaborate with team members by creating branches, reviewing code, and merging changes using pull requests.

### 6. Communication and Collaboration

- Use Discord or Microsoft Teams for team communication, discussions, and updates.

- Utilize email for notifications related to code reviews, pull requests, and project milestones.

## Connecting IntelliJ IDEA to Tomcat Server

1. Open IntelliJ IDEA and navigate to `File > Settings > Build, Execution, Deployment > Application Servers`.

2. Click on the `+` icon and choose `Tomcat Server > Local`.

3. Browse and select the directory where Tomcat Server is installed on your machine.

4. Configure the server settings as needed and click `OK`.

5. Create a new run configuration for your web application by navigating to `Run > Edit Configurations`.

6. Click on the `+` icon and choose `Tomcat Server > Local`.

7. Select the previously configured Tomcat Server and specify the deployment settings for your web application.

8. Save the configuration and run your application by clicking the `Run` button.

## Editing in GitHub

1. Navigate to the GitHub repository of the project.

2. Click on the file you want to edit.

3. Click on the pencil icon in the top right corner to enter edit mode.

4. Make the necessary changes to the file.

5. Add a descriptive commit message describing your changes.

6. Optionally, create a new branch for your changes.

7. Click on `Propose changes` and then `Create pull request` to submit your changes for review.

8. Collaborate with team members through code reviews and discussions.

9. Once approved, merge your changes into the main branch.

## Discord Integration

1. Create a Discord server for your project if you haven't already.

2. Invite team members to join the server.

3. Set up channels for different topics such as general discussion, development updates, and bug tracking.

4. Integrate GitHub with Discord to receive notifications about code changes, pull requests, and other activities.

5. Utilize Discord bots or plugins to automate certain tasks and enhance collaboration.

## Conclusion

This technical documentation provides a comprehensive guide for developing and managing a sporting goods webpage using various tools and technologies. By following the outlined workflow and utilizing the suggested tools, teams can effectively collaborate, develop, and deploy their web application.