

Web Assignment Report

20226758 - Jingqi Fan

July 26, 2024

Abstract

This report presents a JavaServer Faces (JSF) application that combines XHTML, MySQL, and JDBC to enable users to register, log in, and build customizable personal websites. The application integrates a structured database for user profiles and aesthetic choices, such as color schemes and fonts. Users can design their personal pages via an intuitive interface, selecting from multiple page templates that cater to diverse preferences. The system efficiently manages data flow through managed beans, enhancing user interaction and system functionality. This setup provides a user-friendly environment for personal and professional page customization.

1 Back-end Part

1.1 Database Design

The database creates a comprehensive system for managing user profiles, projects, and their associated properties such as collaborators, keywords, color schemes, and fonts, tailored for a personal page application.

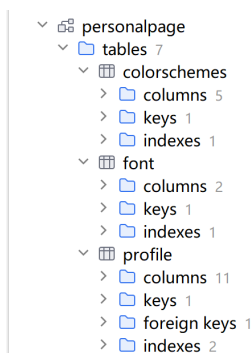


Figure 1: database structure

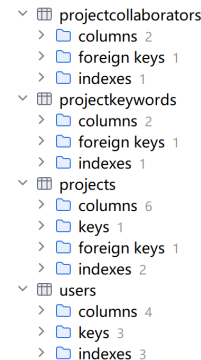


Figure 2: database structure

1.1.1 Table Relationships

1. Projects to Profile

- Foreign Key: profile_id in the Projects table.
- References: id in the profile table.
- Purpose: Links each project to a user's profile, allowing for the association of projects to specific user profiles on the personal page.

2. Project Keywords to Projects:

- Foreign Key: projectId in the ProjectKeywords table.
- References: id in the Projects table.

- Purpose: Establishes a relationship between projects and multiple keywords, enabling the tagging of projects with relevant keywords for better categorization and searchability.
3. Project Collaborators to Projects:
 - Foreign Key: projectId in the ProjectCollaborators table.
 - References: id in the Projects table.
 - Purpose: Links collaborators to specific projects, which helps in identifying all individuals contributing to a project, useful for credit and collaborations.
 4. Profile to ColorSchemes:
 - Foreign Key: color_scheme_id in the profile table.
 - References: id in the ColorSchemes table.
 - Purpose: Allows profiles to specify a color scheme, providing personalized aesthetics for the user's page. This link supports visual customization of the profile.
 5. Profile to Font
 - Foreign Key: font in the profile table.
 - References: id in the Font table.
 - Purpose: Allows profiles to select a font from the predefined options in the Font table.

1.2 Entity and Bean

I complete a well-organized JavaServer Faces (JSF) application that integrates XHTML, MySQL, and JDBC to create a dynamic and interactive personal page system. Here is the structure of entity and beans.

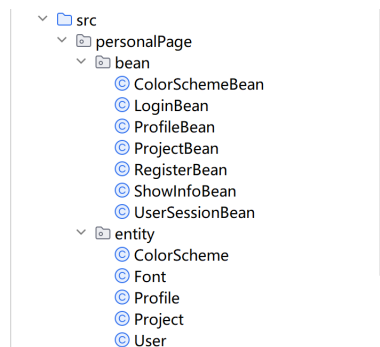


Figure 3: Project Structure

1.2.1 Entity Layer

The entity layer consists of several classes that represent the data model of the application. These classes are directly mapped to the tables in the MySQL database, facilitating data manipulation.

1. User: Represents the users of the application, holding essential data like username, password, and email.
2. Profile: Stores detailed information about the users, such as biography, contact details, and links to social media, making it central to the user's personal page.
3. Project: Contains data related to various projects that a user might want to showcase on their personal page.
4. ColorScheme: Manages different color schemes that users can apply to their personal pages.
5. Font: Handles the font styles available for the user to customize their page text.

1.2.2 Bean Layer

The bean layer features several managed beans that handle the business logic of the application. Managed beans in JSF are used to manage the data flow between the entity classes and the XHTML views.

1. LoginBean: Manages user authentication.
2. RegisterBean: Handles new user registration.
3. ProfileBean: Responsible for retrieving and updating profile information.
4. ProjectBean: Manages project-related operations.
5. ColorSchemeBean: Allows users to select and apply color schemes.
6. ShowInfoBean: Used for displaying user and project information on the personal page.
7. UserSessionBean: Manages user session data across different views and interactions.

2 Front-end Part

The front-end part of the JSF-based application, as depicted in Figures 4 and 5, showcases a structured and methodical approach to handling the user interface components and styling.

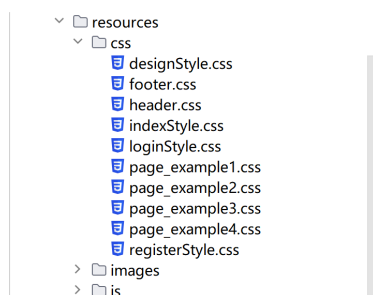


Figure 4: CSS files

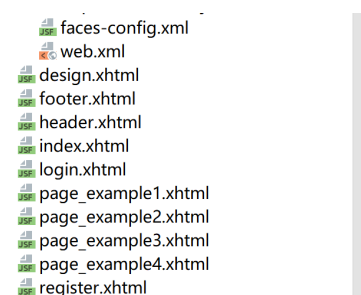


Figure 5: XHTML files

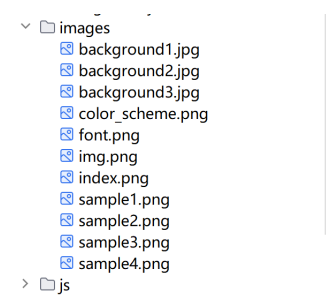


Figure 6: Image files

2.1 Home Page

The Home Page of the application acts as the gateway for new and returning users, providing a clean and intuitive interface that directs users to key functionalities like registration and login. Styled with indexStyle.css, it offers a visually appealing layout with a welcoming introduction in index.xhtml.

2.2 Login And Register

The Login and Register sections feature distinct, user-friendly forms styled with loginStyle.css and registerStyle.css. Users can log in or register through forms that validate inputs and provide immediate feedback.

2.3 Design personal page

In the Design Personal Page section, users are asked to input their informations to create and customize their personal pages with a range of options. Through interactive elements managed by designStyle.css, users can select layouts, color schemes, and fonts to personalize their space. The section is built to support creativity and user engagement.

3 Novelty

3.1 Page Templates

I have designed four unique page templates, each offering a different layout to suit various user needs and aesthetic preferences, thus enhancing the functionality and visual appeal of the personal pages:

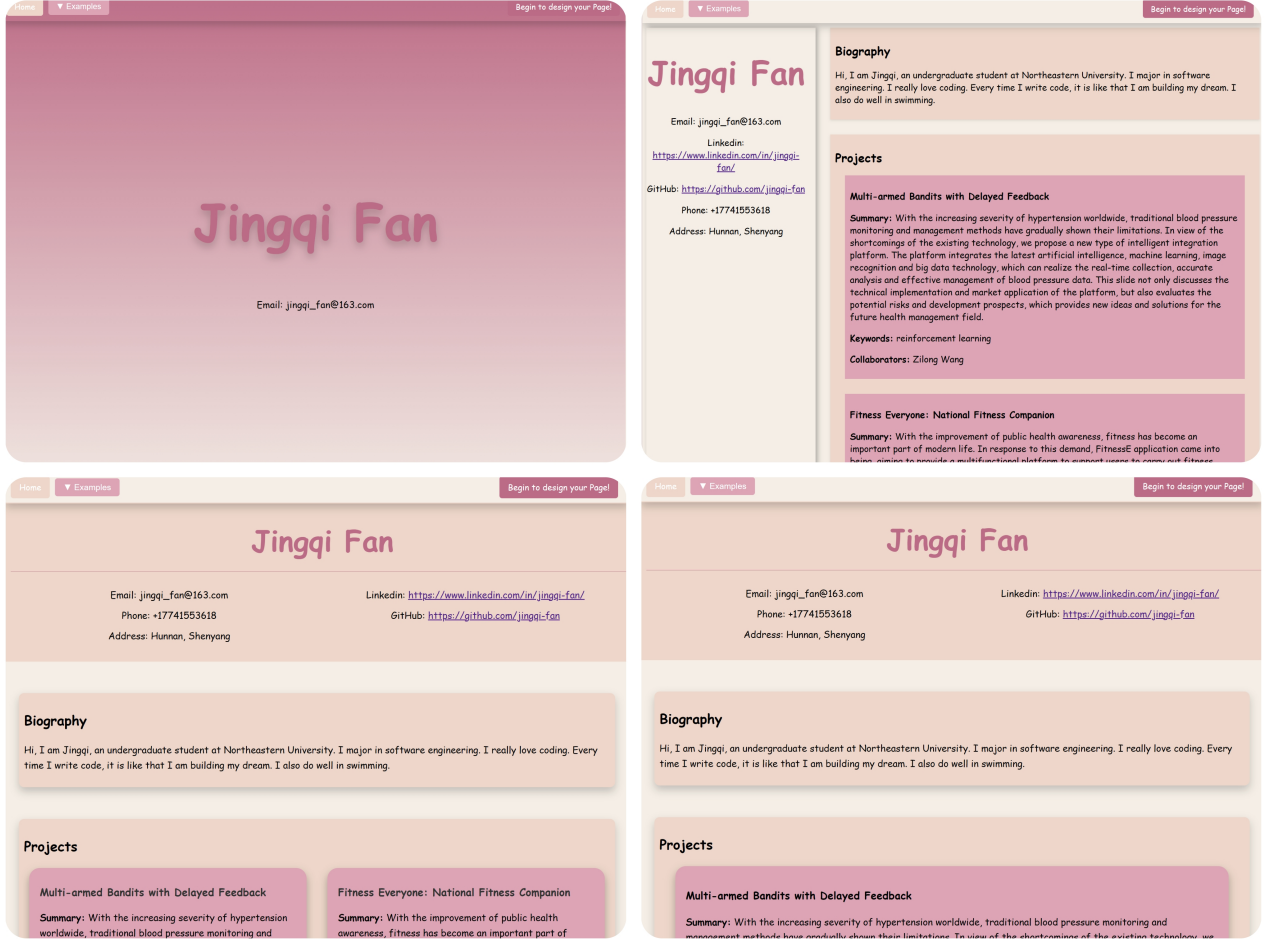


Figure 7: page template

1. Minimalist Layout: This template provides a clean, straightforward presentation of information, enhancing readability and visual appeal.
2. Minimalist Layout with Two-column: Building on the minimalist concept, this layout introduces a two-column format that allows for a structured presentation of projects alongside personal information.
3. Sidebar Layout: This design separates content into a sidebar and a main content area, ensuring that essential personal details are immediately noticeable upon visiting the page.
4. Hero Header Layout: This layout focuses on a large hero header that showcases the user's name or a key visual, creating a strong first impression. Below the header, the page transitions into a more detailed layout.

4 Run this project

The following content is description of the various sections needed to run the project.

4.1 Preparation of Database

Begin by executing the SQL initialization file located at:

```
1 ..\jsf1\WebContent\resources\sql\init.sql
```

This script is essential for setting up the database schema and initializing it with any required data. After running the SQL file, modify the database connection code:

```
1 // Establishes connection to the database
2 private Connection getConnection() throws SQLException, ClassNotFoundException {
3     Class.forName("com.mysql.cj.jdbc.Driver");
4     String url = "jdbc:mysql:///personalpage?useSSL=false&
5         allowPublicKeyRetrieval=true&serverTimezone=UTC";
6     return DriverManager.getConnection(url, "root", "password");
}
```

This involves updating the connection string, username, and password in the Java method that establishes a connection to the MySQL database.

4.2 Libraries

For the project to function correctly, we must import specific libraries that enable JSF capabilities and manage database interactions.

```
1 javaee-api-7.0.jar
2 javax-faces-2.2.8.jar
3 mysql-connector-j-8.3.0.jar
4 primefaces-8.0.jar
```

4.3 Running

Finally, we can run the project in our IDE.

5 Conclusion

This project effectively combines JavaServer Faces, MySQL, and JDBC to create a customizable personal web page system (Wadia et al., 2014). It features a robust database design and user-friendly front-end that allows users to design and personalize their pages with ease. The application's managed beans ensure seamless data integration and user interaction, providing a reliable and scalable platform for personal and professional online presentation.

References

Zubin Wadia, Hazem Saleh, and Allan Christensen. *Pro JSF and HTML5: Building Rich Internet Components*. Apress, 2014.