Table of Contents

USER WEB APPLICATION

Developer: Diemi Pham - HCL America Inc.
Date: December, 2020

Project objective

- **Retrieving the User Details Using the User ID:** Create a servlet-based application that shows a form to enter a user ID. The user ID is then validated, and user details are retrieved from the database and displayed to the user. You need to create a user table in H2 and pre-populate it with data. Use JDBC to do all database processing.
- **Adding a New User in the Database:** Create a servlet-based web application that shows a form to add new users. A H2 database will be created to store user data. The form data will be validated, and a row will be added to the database. All database processing will be done using Hibernate.
- **User Details Portal:** Create a servlet-based web application that shows a form to enter user details. Capture the details in a servlet and then display the data that was entered.
- **Validation of the User Login:** Create a servlet-based web application that shows a login page and validates it. A H2 database will be created to store user data. On successful login, a dashboard page is shown. The dashboard will provide a link for logging out. Incorrect logins need to be handled by showing an error message page.

Background of the problem statement

- **Retrieving the User Details Using the User ID:** As a part of developing an e-commerce web application, the admin backend requires a module that can retrieve user information based on the user ID.
- **Adding a New User in the Database:** As a part of developing an e-commerce web application, you have to create a database table for storing user information. A form is needed to add new users. The form submission is validated, and a new record is created in the user table.
- **User Details Portal:** As a part of developing an ecommerce web application, you have to prototype a form for adding users into the system entered by the users. There is no database involved here, so the product is just captured and displayed without storing it anywhere. (Notes: this requirement is upgraded to store entered information into the database. Hence, this becomes the feature *Adding a New User in the Database* mentioned above.)
- **Validation of the User Login:** As a part of developing an e-commerce web application, you have to prototype a login scenario for the user. A form is needed for users to enter username and password. The form submission is validated, and users are redirected to a welcome page.

Specifications

- **Retrieving the User Details Using the User ID:**
  - There is a JSP page to take in a user ID
  - A servlet that will take the user ID and use JDBC to query the database for the user
  - If the user is found, the servlet will display the user details, otherwise it will show an error message
- **Adding a New User in the Database:**
  - Show an add user form in JSP
  - Handle the form submission in a servlet. Validate it for any missing information
  - Add the user information into the database using Hibernate
  - A Hibernate configuration file is set up using XML for the user table
  - A class is created to store user data. This class will be linked to Hibernate via an hbm.xml file
  - The servlet will put the form data into the user class and then pass the user class into Hibernate to add into the database
- **User Details Portal:** is upgraded to *Adding a New User in the Database*
- **Validation of the User Login:**
  - Show a login form in HTML
  - Handle invalid logins and show appropriate error messages using servlets
  - Show the dashboard page using servlets
  - Handle logouts using servlets

Java Concepts and Data Structures

- H2 database is used for storing user data;
- Servlet and JDBC are used to query the database for the user.
- Hibernate is used for database processing

Repository

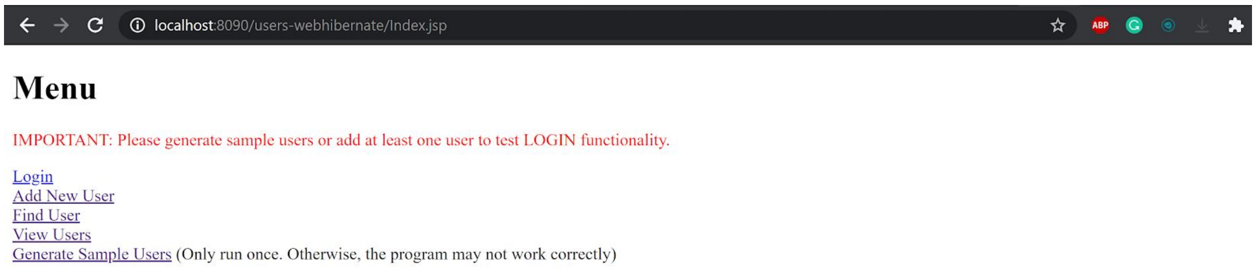
Source Codes
Technical Documentation (online version)

Flow of the application

- **Retrieving the User Details Using the User ID (Find User)**
    - This functionality can be accessed from the index page by clicking on *Find User* hyperlink. The application will redirect to the Find User page (see screenshots sections) and allow users to enter user ID for to lookup. If the input user ID does not exist in the database, a user not found error message will be raised. Otherwise, the application will show the result with the user detail page. The user detail page will not show the password field.
    - From the user details page, also known as *View User* page which can be accessed from the index page, users are able to edit users' information or delete user(s). In edit user form, the user ID cannot be changed.
- **Adding a New User in the Database (Add New User/Register)**
    - Adding or registering a new user can be accessed either from the index page or view users page by clicking on the *Add New User/Register* hyperlink. The application will redirect to the *Add New User/Register* page which contains a form to fill in with user details such as first name, last name, email, and password.
    - By default, the user's email will also be the username for the user to login later, so email and password fields must not be blank. The application also checks if the email input already exists in the database. If so, the application will decline adding/registering the new user. Otherwise, the application will add the new user to the database and show a welcome message to the user.
- **Validation of the User Login (Login)**
    - Users can access the login function by clicking on the *Login* hyperlink from the index page. In the login page, the application asks the user to enter a user and password for verification. By default, username is the user's email. If the credentials are verified successfully, then the application will show a welcome screen with a welcome message and option to logout. Otherwise, a login failed message will be raised.
- **Others**
    - **Index page**: Behaves as a home page of the application. Users start the application by running this index page and access other functionalities from here.
    - **View all users**: Lists all users with details except password. From here, users are able to edit users' information or delete user(s).
    - **Delete user(s)**: Users can delete user records one by one or delete all users at once.
    - **Generate sample users**: Generates sample user records for testing purpose. This function should be run only once. If run more than once, the samples are generated only once due to the uniqueness of username/email inputs.

Screenshots

Main Menu (Index page)



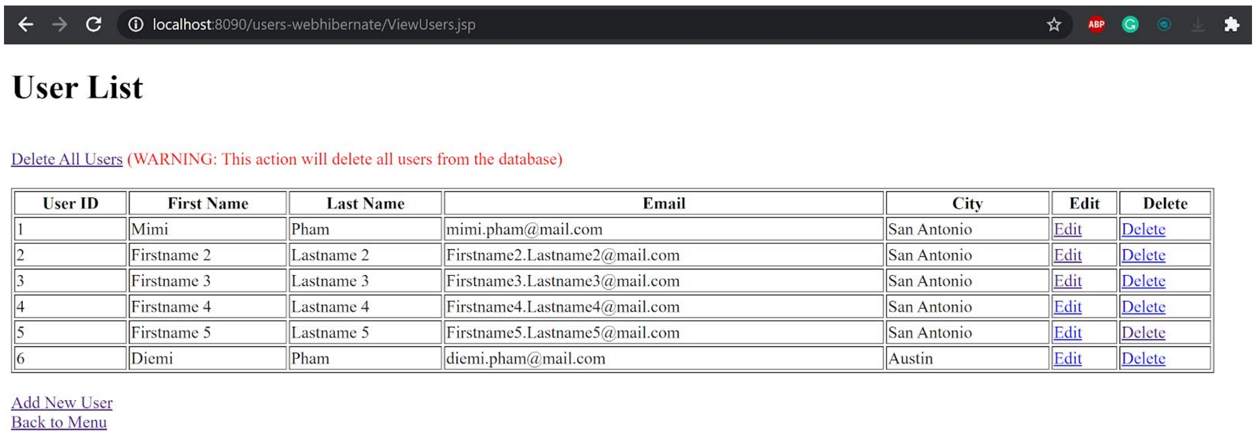
localhost:8090/users-webhibernate/Index.jsp

Menu

IMPORTANT: Please generate sample users or add at least one user to test LOGIN functionality.

Login
Add New User
Find User
View Users
Generate Sample Users (Only run once. Otherwise, the program may not work correctly)

View Users page



localhost:8090/users-webhibernate/ViewUsers.jsp

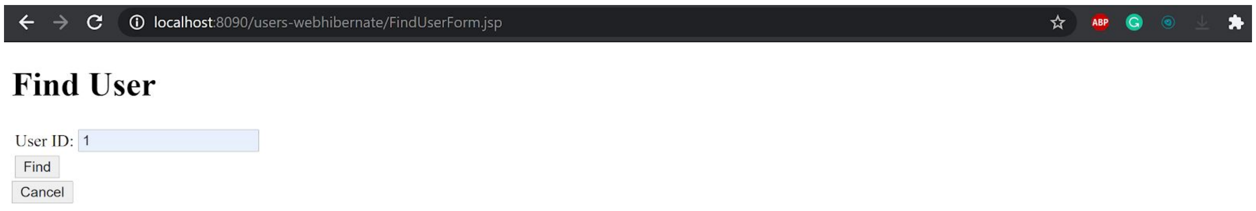
User List

Delete All Users (WARNING: This action will delete all users from the database)

User ID	First Name	Last Name	Email	City	Edit	Delete
1	Mimi	Pham	mimi.pham@mail.com	San Antonio	Edit	Delete
2	Firstname 2	Lastname 2	Firstname2.Lastname2@mail.com	San Antonio	Edit	Delete
3	Firstname 3	Lastname 3	Firstname3.Lastname3@mail.com	San Antonio	Edit	Delete
4	Firstname 4	Lastname 4	Firstname4.Lastname4@mail.com	San Antonio	Edit	Delete
5	Firstname 5	Lastname 5	Firstname5.Lastname5@mail.com	San Antonio	Edit	Delete
6	Diemi	Pham	diemi.pham@mail.com	Austin	Edit	Delete

Add New User
Back to Menu

Find User Page



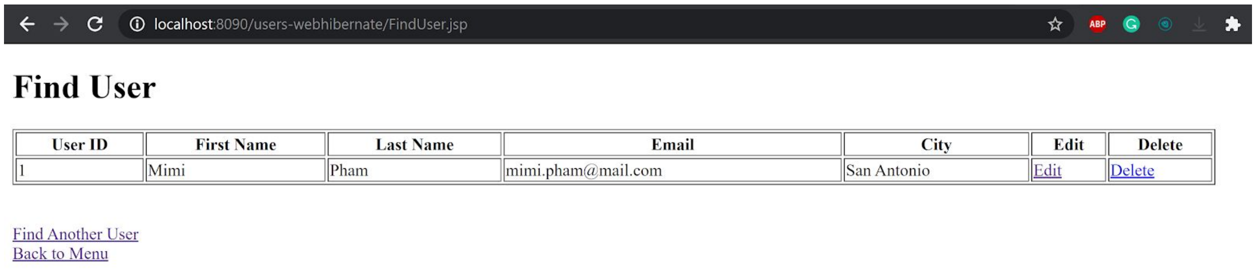
localhost:8090/users-webhibernate/FindUserForm.jsp

Find User

User ID: 1

Find
Cancel

Find User Result - User Found

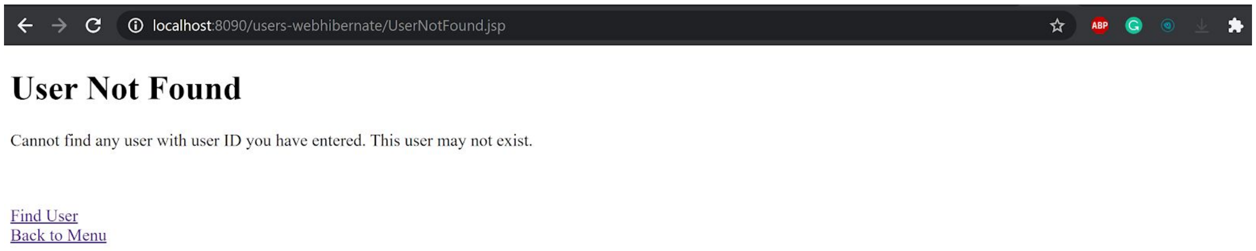

localhost:8090/users-webhibernate/FindUser.jsp

Find User

User ID	First Name	Last Name	Email	City	Edit	Delete
1	Mimi	Pham	mimi.pham@mail.com	San Antonio	Edit	Delete

Find Another User
Back to Menu

Find User Result - User Not Found

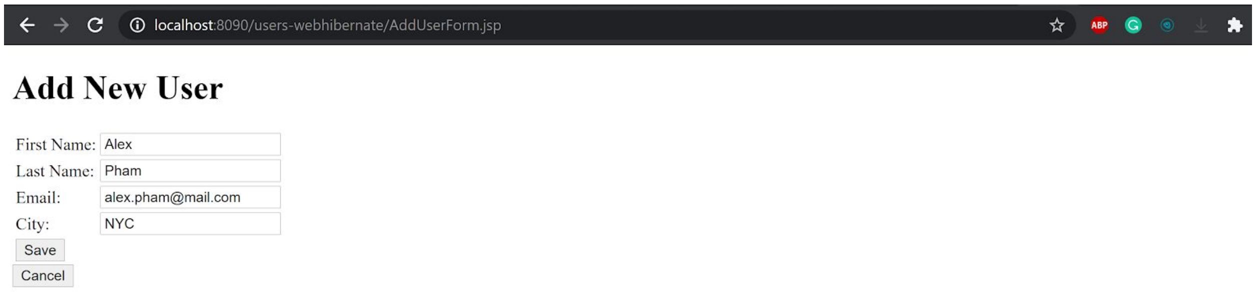

localhost:8090/users-webhibernate/UserNotFound.jsp

User Not Found

Cannot find any user with user ID you have entered. This user may not exist.

Find User
Back to Menu

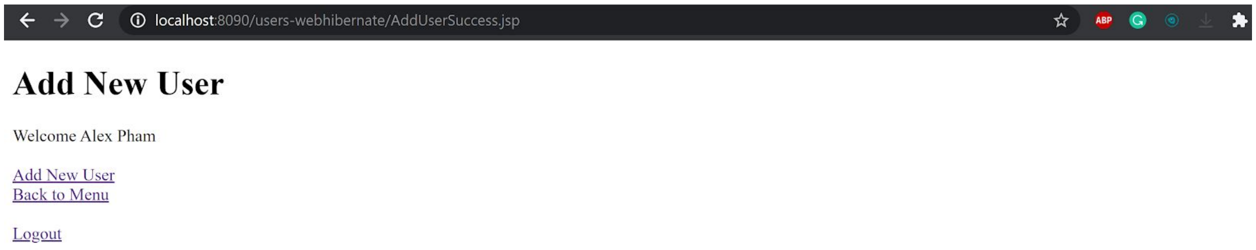
Add New User Page


localhost:8090/users-webhibernate/AddUserForm.jsp

Add New User

First Name:	Alex
Last Name:	Pham
Email:	alex.pham@mail.com
City:	NYC

Add New User Success


localhost:8090/users-webhibernate/AddUserSuccess.jsp

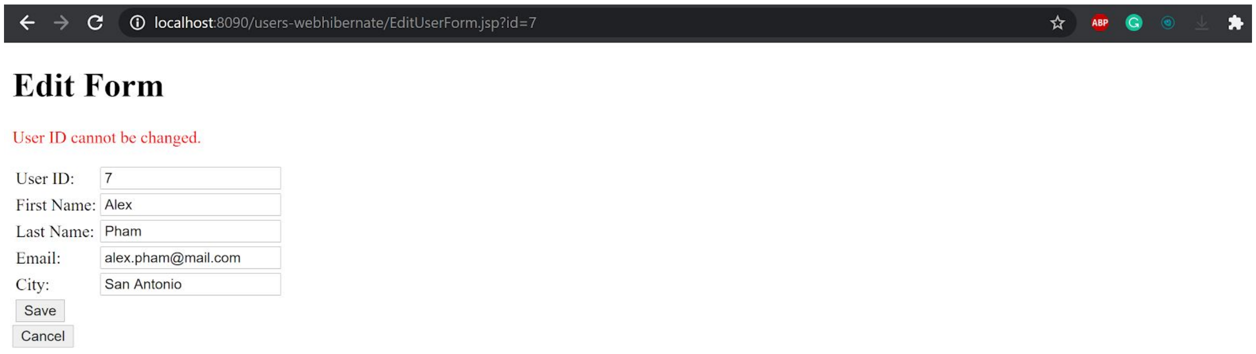
Add New User

Welcome Alex Pham

Add New User
Back to Menu

Logout

Edit User Page

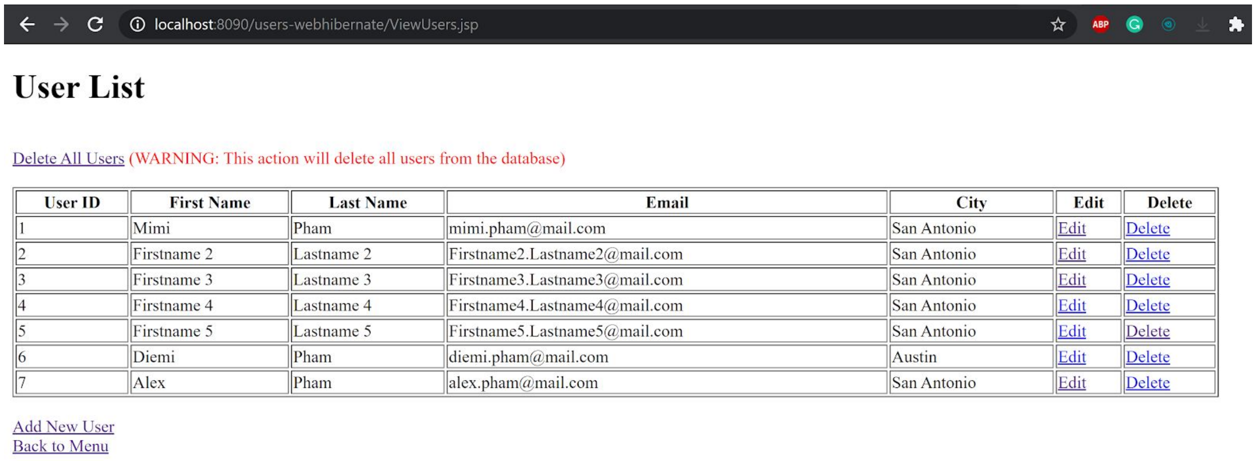

localhost:8090/users-webhibernate/EditUserForm.jsp?id=7

Edit Form

User ID cannot be changed.

User ID:	7
First Name:	Alex
Last Name:	Pham
Email:	alex.pham@mail.com
City:	San Antonio

Edit User Success


localhost:8090/users-webhibernate/ViewUsers.jsp

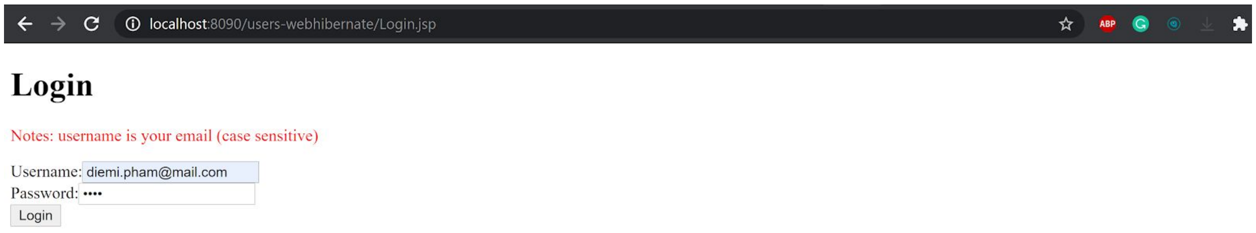
User List

Delete All Users (WARNING: This action will delete all users from the database)

User ID	First Name	Last Name	Email	City	Edit	Delete
1	Mimi	Pham	mimi.pham@mail.com	San Antonio	Edit	Delete
2	Firstname 2	Lastname 2	Firstname2.Lastname2@mail.com	San Antonio	Edit	Delete
3	Firstname 3	Lastname 3	Firstname3.Lastname3@mail.com	San Antonio	Edit	Delete
4	Firstname 4	Lastname 4	Firstname4.Lastname4@mail.com	San Antonio	Edit	Delete
5	Firstname 5	Lastname 5	Firstname5.Lastname5@mail.com	San Antonio	Edit	Delete
6	Diemi	Pham	diemi.pham@mail.com	Austin	Edit	Delete
7	Alex	Pham	alex.pham@mail.com	San Antonio	Edit	Delete

Add New User
Back to Menu

Login page

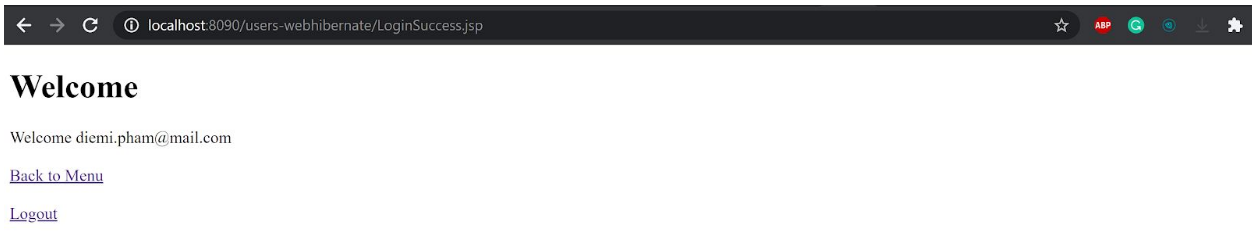

localhost:8090/users-webhibernate/Login.jsp

Login

Notes: username is your email (case sensitive)

Username:	diemi.pham@mail.com
Password:	••••

Login Success



localhost:8090/users-webhibernate/LoginSuccess.jsp

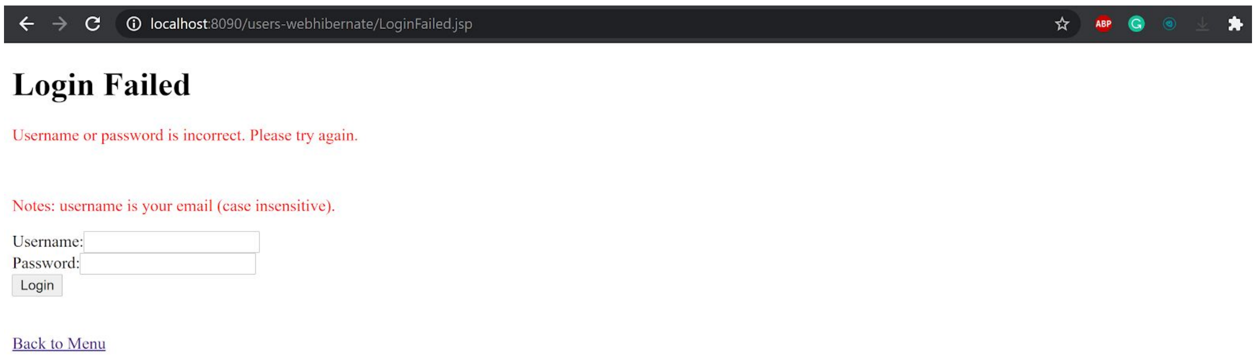
Welcome

Welcome diemi.pham@mail.com

Back to Menu

Logout

Login Failed



localhost:8090/users-webhibernate/LoginFailed.jsp

Login Failed

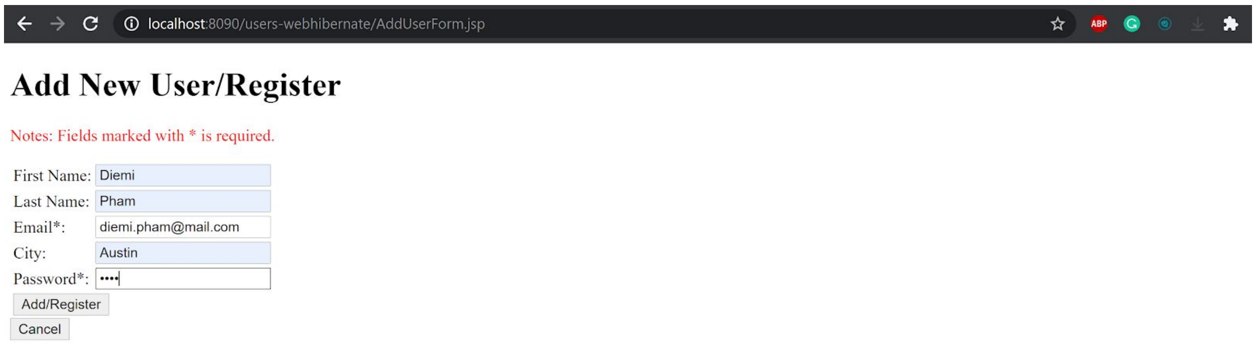
Username or password is incorrect. Please try again.

Notes: username is your email (case insensitive).

Username:
Password:

Back to Menu

Register page



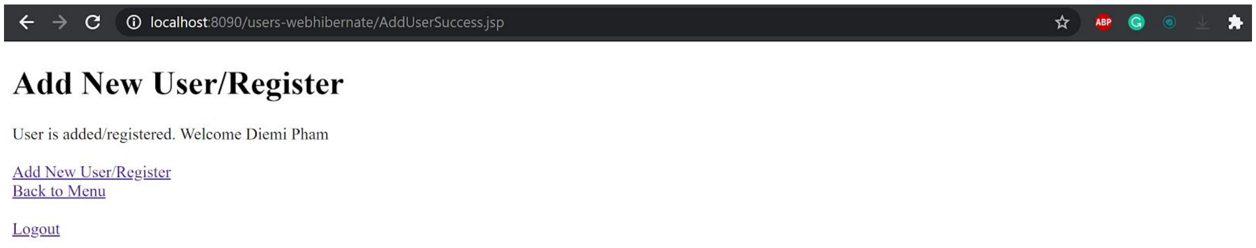
localhost:8090/users-webhibernate/AddUserForm.jsp

Add New User/Register

Notes: Fields marked with * is required.

First Name:	Diemi
Last Name:	Pham
Email*:	diemi.pham@mail.com
City:	Austin
Password*:	••••

Register Success


localhost:8090/users-webhibernate/AddUserSuccess.jsp

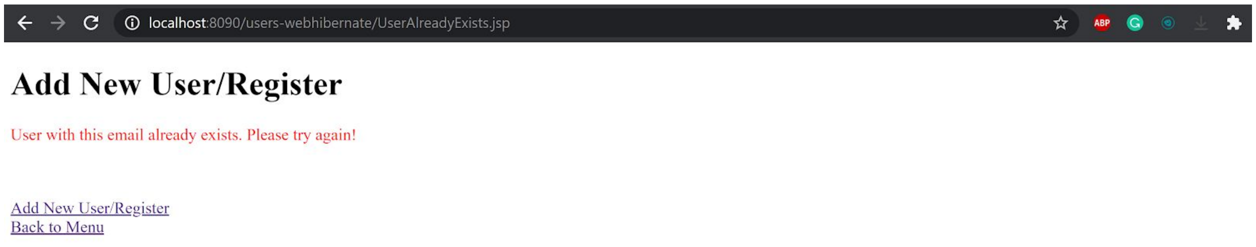
Add New User/Register

User is added/registered. Welcome Diemi Pham

Add New User/Register
Back to Menu

Logout

Register Failed


localhost:8090/users-webhibernate/UserAlreadyExists.jsp

Add New User/Register

User with this email already exists. Please try again!

Add New User/Register
Back to Menu