



Creating a User Profile Page:

User Interface

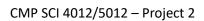
• Design of the UI is up to you. Basic requirement is it supports all the required features and is mobile responsive.

Features

- There are 4 main pages to this experience
 - User Registration Page
 - This is the page where the user can register.
 - Should collect Username, Password, First Name, Last Name, Address Street 1, Address Street 2, City, State, Zip Code.
 - Make a option (like a drop down) to select the form submission type.
 Possible types are GET and POST.
 - o 404 Error Page
 - If the User submits a form submission with GET instead of POST, serve up a custom 404 error page to the user.
 - User Sign In Page
 - If the user already has an account, then let the user sign in without filling out registration again.
 - User Profile Page
 - On this page, the user will be able to add new information and edit some information about this users profile.
 - Allow for adding an image attachment to upload to server. This image would be displayed as the user profile thumbnail picture.
 - Allow for all the information collected at registration (except username) to be edited.
 - Allow for user to enter and update work history, education (current and historical), areas of interest, birthday, phone number, cell phone number, time zone (drop down, perhaps?).

Database

- Exact database implementation is up to each individual. Since we have not covered Hibernate yet, please use JDBC for a database connection to MySQL. I do have some high level suggestions for the database.
- Create MySQL database called "enterprise"
- Create USERS table
 - USERS_ID column, auto increment
 - LOGON ID column, varchar(whatever size you wish)
 - PASSWORD column, varchar(whatever size you wish)
 - o ADDRESS_ID column, foreign key to the ADDERSS table
- Create ADDRESS table





- o ADDRESS ID column, auto increment
- STREET 1 column, varchar(whatever size you wish)
- STREET 2 column, varchar(whatever size you wish)
- CITY column, varchar(whatever size you wish)
- STATE column, varchar(whatever size you wish)
- ZIP_CODE column, varchar(whatever size you wish)
- Create USERDESC table
 - USERS_ID column, foreign key to the USERS table
 - BIRTHDAY column, DATE
 - PHONE column, varchar(whatever size you wish)
 - CELL_PHONE column, varchar(whatever size you wish)
 - TIME_ZONE column, varchar(whatever size you wish)
 - o PROFILE_IMAGE column, longblob
- Create WORKHISTORY table
 - o WORKHISTORY ID column, auto increment
 - o USERS ID column, foreign key to the USERS table
 - JOB_TITLE, varchar(whatever size you wish)
 - o COMPANY NAME, varchar(whatever size you wish)
 - YEARS_OF_SERVICE, number
- Create EDUCATIONHISTORY table
 - o EDUCATIONHISTORY ID column, auto increment
 - o USERS_ID column, foreign key to the USERS table
 - DEGREE_TYPE, varchar(whatever size you wish)
 - DEGREE_DISCIPLINE, varchar(whatever size you wish)
 - o YEAR ACHIEVED, number
 - UNIVERSITY_NAME, varchar(whatever size you wish)

TECHNOLOGY STACK

- JDBC
- JSP
- SPRING MVC (Root Context and then Servlet Contexts)
- Follow best practices

JDBC Information



</dependency>

https://www.tutorialspoint.com/jdbc/jdbc-quick-guide.htm

```
/**
 * @see Servlet#init(ServletConfig)
public void init(ServletConfig config) throws
ServletException {
   // Define connection to database
   try {
      Class.forName("com.mysql.jdbc.Driver");
   } catch (ClassNotFoundException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
   }
   try {
      this.connection = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/enterprise",
"root", "");
   } catch (SQLException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
   super.init(config);
}
```