Network based application development (ITIS 4/5166)

Final Project

For this project, you will deliver a network-based web application.

This is to be completed individually.

The goal for this project if for you to finalize and package the application that we have worked on in the course assignments and use that to build upon as the final project.

The following additional features are required:

1. User Registration
    a. Users should be able to register with a site by creating an account that includes their information.
2. A user that is registered with the site and has an account should have access to the following features:
    a. Swap an item – complete confirm swap functionality
    b. Save item/user feedback – users should be able to provide ratings for their items and items of other users they swapped and the users they swapped with
    c. View / edit items in profile – fix any issues you had from previous assignments requirements.

Project requirements:

1. Use of CSS (do not define styles on each page or each element)
2. Well - documented code
    a. Appropriate use of comments
3. Accessing and writing data to a persistent Database
    a. Permissions appropriate with respect to users
4. Secure Java web application
    a. Application must be secure against known attacks (Cross - site scripting, social engineering)
5. Requirements and Design document
6. User Manual

General Requirements:

Your web application must apply the MVC pattern, according to the following specifications:
1. Data objects to implement the business layer of the application (**model**)
2. Dynamic pages to present the **view** to the browser
3. Controller logic to **control** the flow of the application

Standards for Web Page Design (view):
The following design standards must apply to all pages of your application:
1. Pages must be valid HTML 5 standard. They must validate error - free.
2. All the web pages must be responsive i.e. they should be rendered without distortion across all screen widths (see Responsive Web Design).

3. You must NOT use a site - builder or automatic site generation tools.
4. There must be complete separation of code.
    a. All CSS should be included into separate files – only link(s) to the external CSS files are allowed in the HTML document.
    b. All JavaScript should be included into separate files – only link(s) to the external *.js files are allowed in the HTML document.
5. Each project will consider coding efficiency and design choices, which includes coding elegance and style.
    a. Efficiency will be determined on the use of resources
    b. Code design will be evaluated based on ease of maintenance, understandability, documentation (within code), reusability and extensibility.
        i. Proper indentation will be evaluated
    c. Design will be evaluated based on
        i. Aesthetics - Good visual presentation
        ii. Ease of use
        iii. Good navigation

Standards for Business Logic (control):

Implement controller logic to operationalize the business logic. Examples include:
sending parameters as part of GET/POST http requests

- Populating views with dynamic data
- Controlling navigation within application
- Controlling page and data access based on user credentials

Standards for Business Layer Design (model):

- You will utilize data objects for the model to represent the main data elements/business objects used.

- You will utilize persistent data storage to save application data:
    - Your application must utilize a persistent database that is populated with data
    - Your final submission must include scripts to generate and populate the database tables
    - For Java track - SQL must utilize prepared statements

Requirements and Design Document: (most of this section can be drawn from the assignment descriptions)

1. Project overview: Describe your application. Include what it is supposed to do and who is your target market. Refer to the requirements and address each point.

2. Site Map: Before you start coding, you need to understand what pages you will need to create and how you will navigate between them. A sitemap is list or diagram representing the hierarchical structure of the HTML pages in a website. Please produce a site map for your application.

https://speckyboy.com/collection-inspiring-sitemaps-user-flow-maps/
https://www.gliffy.com/blog/how-to-create-a-sitemap
http://slickplan.com/
https://support.office.com/en-us/article/generate-a-web-site-map-329d4803-f99a-4332-a4db-7ac3fd0d97da
https://support.office.com/en-us/article/take-control-of-your-web-site-with-visio-bae87722-a864-4b88-b684-3acc31f5fdaa

3. Page Design: For **each** page listed in your site map, provide *at least* the following requirements:
   - The name of the page
   - Purpose of the page
   - Audience of page (general users, registered users, etc.)
   - What data (fields) will be presented on the page based on what logic?
   - What validations (if any) will exist on page? (of data fields above)
   - What buttons or hyperlinks will exist, and what actions will be taken when clicked (data saved, navigate to other page, search based on value, etc.)
   - Special notes regarding page (if any)

Note: in some cases, the page displayed might be composed of several underlying pages which should have their own listing (header, footer, navigation bars, etc.). The content of these sub - pages should be presented once. Not repeated on every display page.
If you choose to do so, you can accompany each page with a mock up or prototype that shows how the page will be laid out and even how it might behave. Some of these, like Axure, can even be used to generate your requirements/design content. You can use the following for this:

- http://www.justinmind.com/
- http://www.axure.com/
- https://moqups.com/

4. ER Diagram of persistent database designed.

User Manual:

Here are some steps towards generating an effective, user manual:

1. Define who is your user
2. Write to your user's need
3. Explain the problem the user is trying to solve, then write the solution
4. Format properly
   a. Include title and cover page
   b. Include table of contents
5. Utilize graphic images as needed to support text (screenshots are great illustrations!)

Applying Security Measures:

In the course, we covered main attacks against web applications and described possible counter measures for each of these attacks. The following table summarizes the attacks, their corresponding countermeasures.

Attack	**Prevention**
Cross - site Scripting (XSS)	Input validation and output encoding
SQL injection (Java Track)	Prepared Statements
Social engineering	Hash and Salt Passwords

You may apply any of the utility classes that are provided by the book (e.g. utility class for hashing passwords). Make sure and cite the source of that code in your documentation i.e. comments. Apply the changes to all pages, classes and methods that have the vulnerability.

Final Deliverables:

- Requirements and Design Document
- User Manual – how to use project
- Zip file of final project
- Script to generate and populate database

Grading Criteria:

- Technical (70 pts) (see Project Grading Rubric Document)
- Documentation (30 pts)
    - Project Requirements/Design – 22pts
    - User Manual – 8pts