IAB207: Rapid Web Application Development
Assignment 3 | Due October 31 @ 11:59PM
Developing Online Peer to Peer Marketplace

Overview

The last assignment for this unit is a group assignment in which you will develop a web application that implements the following technologies:

1. HTML & Bootstrap CSS
2. Flask Templates
3. Flask WTForms
4. Flask SQLAlchemy
5. Flask Login

You will form **groups of 3 or 4 students within the same workshop**. Groups with a team size of 3 will need to develop the same functionality as teams with a size of 4. For this assignment, your group will need to nominate an online marketplace designed by one of your team members in the earlier assignments of the unit.

Project Deliverables

Please read the below deliverables (requirements) carefully as every team will be expected to address each of the requirements in their final submission.

You are free to re-use the code from the individual assignments of one of the team members to develop all pages again. Ideally, you may choose to re-use your previous work as it will save you time. As part of the project, the following needs to be developed and demonstrated:

1. A landing page containing some useful message of the website and some recent items that are available for sale. Also, your landing page can also have search functionality with a text-field/drop-down menu that allows users to search by entering the category of the item.

   Note: Text-based search functionality is optional, as it is an advanced feature to use text input to match information about relative items before displaying them. That said, you may alternatively provide a 'search item by category' feature as sufficient minimal viable implementation.

2. A results page that shows the results of a search item by category. The results should be presented in an orderly fashion, using one of the Bootstrap components.

3. An individual item 'details' page, where users should be able to see an item's status information (available or sold). A bidder should be able to bid for items that are marked as

available. The details of the bidding user (including name, email, contact number, and the bid date) must be stored.

Note: To keep it simple, the bid date could be assumed to be the same as the system date requiring no input from the user.

4. A seller who is logged in will have the ability to manage items that they have created. A 'manage items' page relative to sellers will present all the bids and their details (user name, email, phone and bid date) listed to the seller.

5. The seller should be able to settle on a bid, indicate the sale of the item along with the date of sale, thus marking the item as 'sold'.

   Note: To keep it simple requiring no user input, you may assume the date sale to be the same as the system date when it was marked for sale.

6. A page with a form to create an individual item to sell.

7. A user registration page. This page should accept a user's email, password, and username. Validations of input fields are also advised to ensure that malformed input (e.g. an email with a space) are not mistakenly accepted.

8. A login form for users to authenticate, via means of entering their username and password.

9. A separate page where a seller can view past sales showing details of past items, including the relative buyer and sale date.

10. Error Handling:
    a. Handle instances in which a page is not found, and internal server errors within a page that allows users to navigate back to the landing/home page.
    b. A user should be authenticated (via login) to either buy or sell items.
    c. User input validation – empty strings, incorrect input for specific fields, etc. should be correctly managed.

All your pages should include a navigation menu, and should utilise data that is dynamically created and stored in a database solution (use SQLite or PostgreSQL).

Project Management

Teams are required to use a Trello board (www.trello.com) to maintain a list of tasks and ensure that each task is assigned to individual members within the group. Teams are also required to invite their tutor to the team's Trello board, so that your tutor has visibility to your progress and activities.

In the final submission, teams will be required to clearly indicate the contribution of individual members to the overall assessment, which should match the allocations in the Trello board.

In every workshop, teams will be asked to give the tutor an overall update of their project status and discuss any issues or concerns they may have in completing the assessment successfully.

Please note: This will be considered when allocating participation marks.

Development Principles

You must develop code that:

1. Is well written with reasonable comments/documentation.
2. Implements HTML, Bootstrap, and Python Flask (with other Flask modules) only. JavaScript may optionally be used as well.
3. Must be deployed (or show attempt of deployment) on the Heroku Cloud Platform. Instructions to do so are covered in the workshop of week 10.

Database Design

A database solution will be required to achieve desired functionality in your solution. Your database design will consist of four tables:

1. An items table containing the item details – you can add more tables to store more information, or can choose to keep it simple if you see fit.
2. A purchases/orders table storing information about the buyers, sellers, negotiated prices etc.
3. A table to store information about all the bids.
4. A user table storing information about user, including their login credentials.

Things to note

- All code should be written by the team members. Marks will be deducted if automated code generators are used.
- Use of JavaScript is optional. None of the functionality described in this assessment relies on JavaScript, however if you want to support some additional interactive functionality, you are free to do implement it using JavaScript.

Submission Instructions

Cover Page

A cover page needs to be submitted by each group, including each student's name, student ID, and the name of their relative tutor. Two individuals from each team (primary and secondary) must submit the group report on Blackboard. This ensures that even if there is an issue with one submission, that every team will have a backup submission in place.

Source Code

Each team is required to submit a ZIP file containing the following:

1. A 'README' text file that contains the student names and student numbers. If this file is missing, your source code will not be assessed.

2. Relevant directories, python files, HTML files, any static files your web application uses, and the 'README' file.

3. The ZIP file may contain the SQLite database file that you used for testing (optional).

4. Please do not include any **venv** directory (Python virtual environment) of any Python installation. Ensure also that you check your file size before uploading.

5. Report on Heroku Cloud Platform deployment – If you have deployed the application, you must provide the link to the application, and accompanying screenshots of demonstrations. If you were unable to deploy the application on Heroku, you must provide a detailed description of the steps you followed and what failed. Evidence proving that you tried this step should be presented (for example: screenshots of error messages).

Application Demonstration

Each team will demonstrate the application in their respective workshops during Week 13. During your demonstration, you may be asked to provide some development details and your understanding of the frameworks used. You will be given a window of time required to demonstrate qualities respective to the criteria mentioned in the following pages. Through these criteria, your demonstrations will contribute to your final grade for the assessment.

Marking

Web Application Demonstration (30 marks)

Requirements	Fully Functional Requirements are fully met, and the page is fully functional without errors. *A submission of this standard succeeds in the fundamental requirements. The solution is thorough and well tested.*	Incomplete or Errors Present Most of the requirements have been met and the page is mostly functional with a few errors. *In a submission of this standard, there would be evidence of the minimal requirements advised in the project deliverables of the assessment. There may be indications of some errors in the solution, however these would be minimal, and would not impair the usage of the solution.*	Missing Content Requirements misunderstood or ignored, does not meet requirements. *A submission of this standard would not function (or partially function) as a minimal viable solution. There would be presence of serious errors that would impair the solution's basic processes.*
Landing page *(#1)*	3	1.5	0
Search results page *(#2)*	3	1.5	0
Item details page *(#3)*	3	1.5	0
Seller Manage Item page *(#4)*	3	1.5	0
Updating Sale Information *(#5)*	3	1.5	0
Item creation form *(#6)*	3	1.5	0
User registration page *(#7)*	3	1.5	0
User login page *(#8)*	3	1.5	0
Seller past sales page *(#9)*	3	1.5	0
Error handling *(#10)*	3	1.5	0
TOTAL	**30**	**13**	**0**

Development and Deployment (5 marks)

TOTAL: 5 Marks					
Criteria	**5**	**4**	**3**	**2**	**1**
Deployment or attempted deployment on Heroku Cloud Platform (file #5 in the source code zip)	Successfully deployed the application with Flask and PostgreSQL	Successfully deployed the application with Flask and SQLLite	Unsuccessful deployment, application failed to start on Heroku.	Unable to deploy the software on Heroku. There were some missing steps	Unable to use the Heroku interface.

Team Collaboration (15 marks)

TOTAL: 15 Marks					
Criteria	**5**	**4**	**3**	**2**	**1-0**
Team has maintained an up-to-date Trello board to manage the project	Board is well organised, project broken down and assigned to team members	Board is well organised, and broken down to a degree and few tasks assigned to team members	Board is organised and broken down to a degree, with very few tasks assigned to team members	Board is poorly organized broken down to a degree, hard to tell who's doing what	Board is poorly organised, clear evidence that it's not been maintained
Team regularly participated workshops and updates tutors with progress	Participated in all workshops (except for excused absences) and provided clear status updates	Participated in most workshops and provided clear status updates	Participated in some workshops and usually provided status updates	Participated in some workshops and usually provided status update	Barely showed up to workshops or provided updates
Team has completed peer review	Team member has an excellent rating by the team	Team member has an very good rating by the team	Team member has a good rating by the team	Team member has a reasonable rating by the team	Team member has been rated poorly by the team
Please note that peer review is an individual component, that is, your individual marks will depend on how your team has rated you.					