EDMUND RICE COLLEGE – YEAR 12 ASSESSMENT NOTIFICATION

Software Engineering – TASK NO. 3 2025

Assessment Task Date Issued	Wednesday 2nd April
Assessment Task Weighting	30%
Assessment Task Due Date	Term 2 Week 9A Monday 23rd June Period 5

ASSESSMENT OUTCOMES (*The student):*

SE-12-01 justifies methods used to plan, develop and engineer software solutions
SE-12-02 applies structural elements to develop programming code
SE-12-06 justifies the selection and use of tools and resources to design, develop, manage and evaluate software
SE-12-07 designs, develops and implements safe and secure programming solutions
SE-11-08 applies language structures to refine code
SE-11-09 manages and documents the development of a software project

Topics / Content / Skills Assessed	Software Engineering Project: The task will consist of completing the full software development cycle for the students chosen eCommerce site.		
Nature of Task	Students are to complete the software development cycle, recording their design process and create a functioning eCommerce site.		
Specifications	Students are required to submit all components electronically.	Equipment required	N/A
How feedback will be provided	Written ✓ Verbal ✓ Other:		

TASK

The task will consist of students recording their eCommerce site design in the design templates available on google classroom. Once this design is completed, they will implement their site using the Flask framework and a local SQL database. Students should utilise the code from their 'Secure Login' assessment. During normal classes students will be required to check-in with their classroom teacher to review their progress.

Design Phase - Planning Tasks:

a) Start the project plan - carrying out your research, recording your design decisions in the project log.
   Research alternatives to developing the site ourselves.
   Research the Agile, Waterfall and WAgile development approaches - decide which one to adopt.
   Research a suitable installation method: Direct, Phased, Parallel or Pilot
b) Read through the requirements and create flowcharts and algorithms to explain the site processing logic.
   At a minimum the site must consist of the following pages & associated functionality:
   Login, Products, Product_Stock_Level, Shopping Cart, Payment, Order_History
   You must use raw SQL and a local SQL database and one example of using ORM (object relational mapping)
c) Design storyboards for the site, a database schema and a data dictionary

Build Phase - Implementation:

a) Setup source code control
b) Build the eCommerce site using Flask, implementing your flowchart/algorithm logic
   You may utilise the 'Bootstrap' library
c) Test the completed system
d) Present your eCommerce sites features

<u>All code must be original and your own work</u> - other than using the standard python & Flask libraries you MUST NOT COPY code from any other sources. As part of your project submission, you will complete a code interview where you will review your code with your teacher.

Preparing for an assessment: https://bit.ly/3HnUkvP	**Student Post Assessment Task Reflection**: https://bit.ly/3D3PjG2

TASK DETAIL

Design Phase - Planning Tasks:

- Review the system requirements and research eCommerce site ideas.
  Research the development approach and installation method to use - record your findings.
  Decide on an installation method: Direct, Phased, Parallel or Pilot - record your findings.
  Research the Agile, Waterfall and WAgile development approaches - decide on your approach.
  Record all design decisions in the log section of the project plan.
- Decide on your sites UI / UX and functionality - record your design in storyboards.
- Create flowcharts and pseudocode to explain the site's main processing logic.
- Create a Database Schema (include a relational database diagram) and Data Dictionary
  The site must use raw SQL and a local SQL database
  Use one example of using ORM (object relational mapping) as covered in class
- Make a task list to complete the site, record in the template Gantt chart provided.

Build Phase - Implementation:

- Setup source code control using git/gitHub.
- Build your eCommerce site using Flask, implementing your flowchart/algorithm logic.
  Reuse the code from your 'Secure Login', implementing the associated security mechanisms
  Follow all python coding conventions (such as following intrinsic variable naming conventions, modular design, exception handling) and provide appropriate source code documentation
  Document one example of code optomisation in your project.
- Test the completed system providing evidence of using a suitable number of 'test cases' (including any white/grey/black box analysis), 'boundary' values and consideration of automated test tools
- Final eCommerce site presentation

Reference Sources:

- Class notes/coding templates & examples from our Google classroom
- EdStem and Eforge online course material
- Your own sources

MARKING CRITERIA - Total 36 marks

eCommerce Site(30 marks):

25-30	- All Project Plan components are completed to an Extensive level (Project Log of all design decisions, Gantt Chart, Storyboards, Database Schema, Data Dictionary and Algorithms, Testing). - The eCommerce site is completed with an extensive level of functionality for the required Login, Products, Product_Stock_Level, Shopping_Cart, Payment and Order_History. - Innovative use of UI/UX features, showcasing an appropriate consistent interface. - An extensive level of Flask/python coding practice - highlighting the use of a robust modular design.
19-24	- All Project Plan components are completed to a thorough level (Project Log of all design decisions, Gantt Chart, Storyboards, Database Schema, Data Dictionary and Algorithms, Testing). - The eCommerce site is completed with a thorough level of functionality for the required Login, Products, Product_Stock_Level, Shopping_Cart, Payment and Order_History. - Use of a range of UI/UX features, showcasing an appropriate consistent interface. - A thorough level of Flask/python coding practice - highlighting the use of a robust modular design.
13-18	- All Project Plan components are completed to a sound level (Project Log of design decisions, Gantt Chart, Storyboards, Database Schema, Data Dictionary and Algorithms, Testing). - The eCommerce site is completed with a sound level of functionality for the required Login, Products, Product_Stock_Level, Shopping_Cart, Payment and Order_History. - Use of UI/UX features, showcasing an appropriate interface. - A sound level of Flask/python coding practice is displayed.
7-12	- The main Project Plan components are completed to a basic level. - The eCommerce site is completed with a basic level of functionality for the required pages. - An appropriate interface is presented. - A basic level of Flask/python coding is displayed.
0-6	- The main Project Plan components are completed to an elementary level. - The eCommerce site is incomplete and/or with an elementary level of functionality for some of the pages. - An elementary level interface is presented. - An elementary level of Flask/python coding is displayed.

- **Code Interview** (**6 marks)**)

5-6	All design decisions are justified fully, with any randomly selected sections of code fully explained with regard to the design decisions involved. The student can highlight the flask/python/html language language features used and explain how they would modify their code to answer a series of suggested updates.
3-4	Some design decisions are justified, with some selected sections of code partially explained with regard to the decisions involved. The student can recognise the flask/python/html language features used and suggest options on how to modify their code to answer a suggested update.
1-2	Some sections of code are explained with regard to the language features used and the underlying design decisions involved.

MARKING CRITERIA - Further breakdown for eCommerce site

- All Project Plan components are completed to an Extensive level (Project Log of all design decisions, Gantt Chart, Storyboards, Database Schema, Data Dictionary and Algorithms, Testing).
    - Project Log contains design decisions, time-stamped accurate and extensive level of all project decisions (design and implementation)
    - Gantt Chart - all distinct tasks are present and in a logical order
    - Storyboards - complete site map and all page components are present in detail
    - Database Schema - with all key(primary/foreign) information, logical and efficient table design
    - Algorithms - main processing loops (as flowchart or pseudocode) accurately reflecting actual implementation
    - Testing - test cases (including any white/grey/black box analysis), 'boundary' values and consideration of automated test tools

- The eCommerce site is completed with an extensive level of functionality for:
    - Login/Manage User - robust, containing user/billing/address information
    - Products - containing an image, cost, description
    - Product_Stock_Level - amount of each product currently 'in stock', changed by completed orders
    - Shopping_Cart - products with a quantity, total
    - Payment - generated receipt with address information, quantity, payment correct
    - Order_History - for a given user, correct previous completed orders are displayed

- Innovative use of UI/UX features, showcasing an appropriate consistent interface
    - Range of appropriate interface features (Bootstrap can be used) showcasing UI components

- An extensive level of Flask/python coding practice - highlighting the use of a robust modular design
    - Modular python code implemented in a robust manner
    - Extensive level of robust python/flask code demonstrated, exception handling/etc implemented

Implementation Scaffold

Get Started	*Review the notification, highlight the important key points* Start the Project Log - record all your design decisions Complete the 'INTRODUCTION' section of the 'Project Plan Conduct your research of Development Approaches (Agile, Waterfall, WAgile) & Installation Methods (Direct, Phased, Parallel or Pilot) Document your chosen Development Approach and Installation Method.
Design	*Add your design decisions into the logfile, update as you progress:* Experiment with the Cart, etc. templates provided on classroom and the bootstrap library Create a Gantt Chart outlining the project time-line Document your site design using Storyboards Decide on what tables you need - create a Database Schema Create a data dictionary Create a Flowchart for the main Flask processing routines
Implementation	*Add your implementation/code decisions into the logfile, update as you progress:* Use your storyboards to design the UI, ensuring a consistent interface Use the pseudocode/flowchart to code the main processing routines Create Flask served pages for: Login, Products, Product_Stock_Level, Shopping_Cart, Payment and Order_History Provide an example of code optomisation
Testing	*Add your testing decisions into the logfile* Test the completed system, document your use of 'test cases' / boundary values / white/grey/black box analysis Give consideration to the use of automated testing tools