IS2108 – Full-stack Software Engineering for AI Solutions I

AY 2025/26 Semester 1

Pair Project

Learning Objectives

At the completion of the pair project, you should:

1. Develop a better understanding of digital commerce from a business-to-consumer (B2C) perspective.

2. Acquire skills to develop a complete B2C e-commerce web application using Python and Django conforming to best practices.

3. Understand how to integrate AI and machine learning models into a software application to automate business processes and decision making.

Opening Narrative – AuroraMart – Personalised E-commerce

Who is AuroraMart?

AuroraMart

AuroraMart is a venture-backed B2C e-commerce start-up aimed at the mass-affluent and value-conscious shopper in Southeast Asia. The product catalogue spans Electronics, Fashion (Men & Women), Home & Kitchen, Beauty & Personal Care, Sports & Outdoors, Books, Groceries & Gourmet, and more, curated to offer breadth without decision fatigue. The brand voice is practical, upbeat, and trustworthy: think "everyday quality without the luxury markup."

AuroraMart's pitch to investors is simple: win the first session, keep the second. That means two things:

- A shopper should find something relevant within the first 90 seconds; and

- The experience should be sticky enough to bring them back within 14–21 days.

Your group have been given six weeks to deliver a demo-ready, data-infused storefront that proves the above concept can actually work at scale.

Market Situation & Competitive Pressure

The B2C e-commerce market in Southeast Asia is characterized by a hypercompetitive environment:

- Crowded field: Regional marketplaces (mass scale, low curation) vs. speciality boutiques (high curation, low scale).
- Differentiation: AuroraMart competes on guided discovery—turning a sprawl of SKUs into personalised storefronts.
- Macro constraints: Rising logistics costs and tighter consumer budgets push the business to focus on attach rate (cross-sell/upsell) and repeat purchase interval (groceries, beauty, consumables).

Core Business Hypothesis

AuroraMart's aims to leverage on the power of data and AI/machine learning. If the company can personalise the first session with a low-friction preference model and use in-basket, high-confidence association rules to encourage add-ons, it will be able to improve conversion, AOV (Average Orde Value), and repeat purchase without heavy discounts.

Decision Tree Classification (cold-start personalisation):

- Input: a handful of onboarding fields (age, gender, employment, income range).
- Output: predicted preferred category → land user on a curated web page immediately.
- Why a tree? Fast, explainable ("We recommended Sports & Outdoors because…").

Association Rules Mining (tasteful selling, not shouting):

- "Frequently bought together" on product pages.
- "Complete the set" in cart based on the current basket.
- "Next best action" in category lists to nudge exploration.

Datasets

AuroraMart's parent, The Merlion Group, presently operates a traditional brick-and-mortar hypermarket in Singapore. Using point-of-sale and customer loyalty data from the hypermarket, a collection of datasets has been created to help implement AuroraMart's personalised e-commerce strategy.

- **Product Catalogue (500 SKUs)** – SKU, name, description, category/subcategory, price, rating, stock, reorder threshold. This dataset is to be treated as the single source of truth for merchandising and availability.

- **Customer Profiles (100 rows)** – Demographics and socio-economics information for 100 customers with labelled preferred category. This dataset was used to train and validate a decision tree classification model for predicting the preferred product category of new customers.

- **Transactions (50,000 baskets × 500 items)** – Actual shopping data of customers. This dataset was used to extract a set of strong association rules that can be used for making product recommendations. For example, a rule that says customer who have purchased product A and B will also buy product C. If a customer adds product A and B to his/her shopping cart, product C can be recommended at the checkout page.

Project Overview

General Requirements

You are required to form a 2-member group to undertake the project and deliver a fully functional prototype of AuroraMart's B2C e-commerce web application.

The project is broadly divided into two different phases – a **design phase** and then follows by an **implementation phase**.

- **Design Phase** – Conceptualise the project backlog or user stories for the web application and design its user interface/user experience (UI/UX). For the latter, you are encouraged to create some wireframes. You are also required to design a suitable relational data model for the web application. It is not necessary to draw an entity-relationship diagram (ERD) or UML class diagram.

- **Implementation Phase** – Develop the web application and implement all user stories created during the design phase.

  The web application should consist of an admin panel and online storefront. The admin panel is for employees to manage the product catalogue, inventory data and customer records. The online storefront is for customers to shop for products.

Deliverables

Your group is required to prepare a **written report** to convey the design of the web application. The written report must contain the following contents:

1. Cover page (include group number, and full name and student number of all group members).

2. Table of contents.

3. **User stories** – User stories for admin panel and online storefront that capture the descriptions of desired features from the first party perspective. User stories should provide simplified and understandable descriptions of business requirements.

For each user story, provide the following details:

- Story Title/ID
- Story description comprising **role**, **goal** and **benefit**:
  - Example – As a Netflix subscriber, I would like the ability to download shows and movies so I can watch them in areas with no or poor network connectivity.
- Notes or any additional context
- Wireframes and/or actual screenshots

4. **Use of Generative AI tools** – Explain how your group have used generative AI tools in the project. Provide details on the tools, processes and evidence (e.g., prompts, conversation history, screenshots, etc.). Reflect on the use of generative AI tools, commenting on the advantages and disadvantages.

5. Appendices (if any)

Please format the written report based on the following settings:

- Paper size: A4
- Margins: 1 inch all around.
- Orientation: Portrait.
- Line Spacing: Double spacing.
- Font Type for Main Text: Times New Roman.
- Font Size for Main Text: 12 pt.
- Page Limit: There is no page limit.
- File Format: DOCX. Do not convert the document into PDF.
- File Name: Name the file as "**PPXX.docx**" where "PPXX" is your group number that will be given to you during registration. If your group number is "PP01", please name your document as "PP01.docx".

The **project source code** should be placed in a single zip archive file with the following folder structure:

- **docs** subfolder containing:
  - A softcopy of your written report in Microsoft Word DOCX format named with your group number.
- **source** subfolder containing:
  - The parent folder of the Django project.
  - Exclude the .git folder and venv folder.
  - Include requirements.txt
- **readme.txt** containing:
  - Group number.
  - Full name, student number, email and contact number of all group members. Please indicate the group leader.
  - YouTube link to your group's full demo video.

Project presentation will be done via a pre-recorded video with a maximum duration of 15 minutes. You are only required to demonstrate the user stories for the admin panel and online storefront in the video. It is not necessary to prepare a presentation slide deck. The video can be recorded with any tool and in any format. All group members should be involved in the presentation. Upload the video to YouTube and provide the link in the readme.txt of your source code deliverable.

Overall Assessment Criteria

Written Report • Appropriateness and completeness of user stories. • Quality of user stories. • Reflection on use of generative AI tools.	**7%**
Implementation • Quality of user stories for admin panel. • Quality of user stories for online storefront. • Appropriateness and completeness of data model. • Integration of AI and ML models.	**30%**
Presentation • Quality of the demonstration.	**3%**
Total	**40%**

Detailed Technical Requirements

Development Stack

You are required to develop the web application in Python using the Django framework with a suitable relational database. By default, Django uses SQLite, which is acceptable. It is not necessary to use a server-based relational database management system. However, the use of object-relational mapping with Django models is mandatory.

Software Engineering Requirements

Your group is required to conceptualise the functional requirements of the web application and create the necessary user stories.

The admin panel must provide functionalities for employees to manage the product catalogue, inventory data and customer records. Handling of order fulfillment is optional.

The online storefront is for customers to shop for products and should feature an intuitive shopping experience. Handling of order tracking, review and rating as well as other after-sale processes is optional.

AI and Machine Learning Requirements

You are provided with two pretrained machine learning models persisted in the `joblib` format:

- Decision tree classification model for predicting the preferred product category of a customer.
- Association rules for recommending products to purchase.

Each model comes with a Notebook that includes details on model training, model persistence and model scoring. Study the Notebooks carefully to understand how to use them. You may treat the models as black boxes and it is not necessary for you to understand their underlying working principles.

You are required to integrate the two models seamlessly into the web application with appropriate user stories. These user stories must directly utilise the models' output for automating some business processes or decision making.

Use of Generative AI Tools

Your group is allowed to use generative AI tools for the project. However, you are not allowed to use a generative AI tool to outright generate the entire codebase of the web application. Such use of generative AI tools will be considered as an act of academic dishonesty and subjected to disciplinary actions.

You are only allowed to apply human-in-the-loop AI-assisted coding. Such use of generative AI tools must be documented in the written report with appropriate evidence. Failure to declare the use of generative AI tools or provide evidence will also be considered as an act of academic dishonesty.

Detailed Assessment Criteria for Project Implementation

The software engineering and AI/ML requirements of the user stories will be assessed via the video demonstration and code review based on the following criteria:

Software Engineering – Functional Breadth Sufficiency of the actual user stories implemented relative to a typical B2C e-commerce business model.	30%
Software Engineering – Functional Depth Relative complexity of the actual user stories implemented.	20%
AI and Machine Learning Appropriateness and correctness of model integration as well as relative complexity of the associated user stories.	20%
User Interface – Functional Aspects Ease of interactivity and navigability of the web application, including the use of appropriate UI/UX principles.	20%
User Interface – Aesthetic Aspects Aesthetic look and feel of the web application.	10%
Total (30% for implementation)	**100%**

Plagiarism and Academic Honesty

The University takes a serious view against any forms of plagiarism. Please refer to the NUS Plagiarism Notice (http://www.comp.nus.edu.sg/cug/plagiarism) for more information. All submitted deliverables will be subjected to a plagiarism check.

You may discuss with other students on specific project tasks, but the final deliverables submitted must be your group's original work.

Once again, you are reminded that the undeclared use of generative AI tools or lack of documentation in the written report is considered an act of academic dishonesty.

Any group or individual student who is found guilty of committing an act of academic dishonesty, e.g., plagiarism, soliciting external help or undeclared use of generative AI tools, will be severely dealt with. Your group or you may be given a **FAIL** grade.

-- End of Pair Project Specification --