UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS 221 Practical Assignment 5

- Date Issued: **23 April 2025**
- Date Due: **27 May 2025**
- Demo Date: **28 May 2025**
- Submission Procedure: **Upload to ClickUP**
- This assignment consists of **10 tasks** for a total of **260 marks**.
- There are **10 marks** assigned to group cohesion for this practical assignment.

1 Overview

This last practical assignment in COS221 serves to expose you to projects and activities that will be required of you in the second semester of your second year as well as on the third-year level, especially COS301. This project is a group-based project which will require you to use all of your obtained knowledge in COS221 plus the effort of your group to complete this assignment. This assignment has minimal specifications to allow you to implement all you have learned, but also discover and showcase to the lecturers and yourself what you are capable of achieving as a Computer Science student of the University of Pretoria.

For this assignment, you are welcome to do and implement everything that you think is suitable for this project. This means that you are welcome to do more than what is required, but not less. Where specifications might seem vague to yourself or your group, make the required decisions and assumptions, and add them to your uploaded PDF. Simply put, you are required to use your creativity for this assignment, and working in groups will make it easier to find ideas and implement a good project. This process is your first introduction to what will be required of you in your final year and industry as a University of Pretoria graduate.

2 Project Scenario

The world of online shopping is vast and ever-growing, with consumers often overwhelmed by the amount of options and fluctuating prices across platforms. Over the years, platforms like PriceCheck have attempted to centralise price comparisons across multiple stores, but many of these services have seen a decline in popularity due to clunky interfaces, outdated pricing, lack of community interaction, and poor user experience.

Our fictional platform, CompareIt, aims to change that narrative. Your group has been commissioned to design and develop a next-generation price comparison web application that not only aggregates up-to-date price listings from various online and physical retailers globally, but also introduces a clean, user-friendly experience that today's consumers expect.

In addition to displaying item prices from multiple stores, the platform should allow users to:

- Browse product categories
- Filter based on criteria such as brand, stockist, user rating or any other criteria
- View a detailed product description with specifications, images, and more
- See a dynamic list of retailers offering the product, along with their pricing

- Leave product reviews and rate items based on their experiences

For this assignment, you can make use of the data that is provided below, but it is not required. You're encouraged to augment this data with realistic mock data or integrate external APIs to enrich the user experience.

- `https://dummyjson.com/products`
- `https://fakestoreapi.com/`
- `https://serpapi.com/`
- `https://www.kaggle.com/datasets/elvinrustam/electronics-dataset`

3 Outcomes

After successful completion of this assignment you should be able to:

- Analyse and understand data from multiple sources
- Able to curate data
- Design a database schema to be implemented in a RDBMS for the curated data
- Design and build a web-based application and:
  - Be able to execute a connection to an RDBMS from a programming language
  - Query and manipulate a relational database from a programming language
  - Build a Graphical User Interface (GUI)
  - Utilise the GUI to query and manipulate a relational database.

4 Constraints

1. Utilise the GUI to query and manipulate a relational database.
2. You may ask the Teaching Assistants for help but they will not be able to give you the solutions.
3. *For the design task and the relational model, you are required to use draw.io* `https://app.diagrams.net/` *or* `https://draw.io/` *or a similar software to create your design. Unless explicitly stated otherwise, use* ***Chen's notation*** *for ER and (E)ER modeling. Failure to adhere to these requirements may result in marks being deducted. This diagram must be in the submitted document for the project*
4. Hand-drawn diagrams will not be accepted. Diagrams must be created using appropriate diagramming software.
5. The PDF, database dump, source code, GUI, and git history will be marked.
6. The GUI interfaces:
   (a) Which run and perform what they are supposed to do get full marks
   (b) Which run but do not perform as required, will receive partial marks
   (c) Which do not run will be allocated partial marks based on the functionality they would have exhibited.
7. You need to use a RDBMS and tools/languages you require to build a web-based application to complete the practical assignment.
8. You may utilise any text editor or IDE, upon an OS of your choice.
9. The group will be required to demo the project; failure to demo and adhere to the guidelines set out may lead to penalties or a mark of zero.

5 Milestones

To be able to successfully deliver and demonstrate the project on 28 May 2025, you need to adhere to the following milestones:

- Register your teams from 23 April 2025 to 27 April 2025
- Initial approach and ideas for the project by 30 April 2025. (To be presented and discussed with your appointed tutors by appointment during the week of 30 April)
- Show your database with valid data and some queries to your tutor on 14 May 2025.
- Demonstrate your initial web-based application that integrates with the database, which represents your minimal viable product on 21 May 2025.
- Bookings for the team demonstration will open on 22 May 2025 at 11:00 and bookings will close on 23 May at 11:00.
- Demonstrate your final project on 28 May 2025.

6 Submission Instructions

You are required to upload a single archive that includes the following files:

- An archive containing your web-based application. If you used a management tool you are required to mention it.
- A pdf containing the answers to the tasks. As well as the link to your GitHub Repository.
- A file or files containing the SQL statements from your database dump to:
  - Create your database,
  - Create the tables in your database
  - Populate the tables with the data you populated the tables with.
- Your archive containing your .git folder
- A **readme.txt** file or a **readme.md** file (which is also used on GitHub) informing the marker what they should do to build and execute your application.

Upload your archive to ClickUP. No late submissions will be accepted, so make sure you upload in good time.

7 Online Resources

- Git: `https://git-scm.com`
- Git Helpful Website with commands and more info on Git: `https://www.atlassian.com/git`
- GitHub: `https://github.com`
- PHP: `https://www.php.net`
- MariaDB: `https://mariadb.com`
- Composer: `https://getcomposer.org`
- Getting Started with MariaDB at: `https://mariadb.com/get-started-with-mariadb/`
- Platform for developers to learn, share knowledge, and build a career: `https://stackoverflow.com/`

8 Assignment Instructions

Task 1: Research ..... (20 marks)

Conduct research on the retail and e-commerce industry, focusing on online shopping and price comparisons. Include references (using IEEE or ACM format) and limit your research to one page, the references can continue onto another page. Cover aspects such as products bought by consumers, price comparison tools, popular product categories, and user experience importance.

Please note that Task 1 must be submitted by **2 May 2025** and will be marked separately. However, it is still a required component of the final project submission.

Task 2: (E)ER-Diagram ..... (30 marks)

After you have conducted your research in Task 1, you are now required to construct an (E)ER-diagram of your project and provide the final model in your uploaded PDF. You are required to mention all assumptions you have made during your modeling and any other information you deem necessary. If you made multiple iterations to get to your final (E)ER-diagram, be sure to include all iterations, as well as notes on how each iteration improves on the previous.

Task 3: (E)ER-diagram to Relational Mapping ..... (30 marks)

Apply the steps for converting your (E)ER-diagram into a relational model. Provide a relational mapping of your (E)ER-diagram in your uploaded PDF. Be sure to indicate the conversion for each step, as well as the assumptions you have made and the choice you have made if the conversion could provide multiple solutions.

Task 4: Relational Schema ..... (30 marks)

Design a relational database schema based on your mappings obtained in Task 3 in the form of both a visual diagram and SQL statements targeted towards MariaDB. Ensure that you include, where applicable, all of the following:

- Primary, Secondary and Foreign Keys
- Constraints and Checks
- Data Types and Length Constraints

Task 5: Web-based Application ..... (55 marks)

In COS216 you learned how to create a web application using PHP, HTML5, CSS, and JavaScript. For this task, you are required to implement a web application to manage and perform different queries you find necessary for this project. Your web application must at a minimum be able to:

- Login and manage users
- Add, edit, and delete products and categories
- Manage stockists and/or e-commerce websites that are stockists, product categories, product descriptions, product prices, product images, and brands
- Sort and filter products based on various criteria
- Update the database and/or extend, delete, etc.
- View a product, and display an image of the product, the list of prices at different retailers (stockists), ratings and/or reviews.
- Allow users to rate and/or review products
- A dashboard, that shows top-rated products, and review data visualisation
- SQL injection prevention

Task 6: Data ..... (20 marks)

To ensure your web application has sufficient data for testing and demonstration purposes, you are required to populate your database with a substantial amount of products, retailers(stockists), brands, categories, and user reviews.

You are welcome to augment the given data and/or include all your relevant data in your database by hand, by using a script that generates artificial data, a script that parses data from an API, or by loading data using XML/JSON feeds from relevant websites or APIs (that you find or write yourself).

You must explain your data population method and the reasoning behind your choices in your uploaded PDF.

Task 7: Analyse and Optimise ..... (10 marks)

You are required to make use of your chosen RDBMS and tools to analyse at least one of your query execution plans and report on the performance to execute provided queries in your uploaded PDF. Thereafter you are required to explain how you would optimise your query in your PDF, implement your proposed optimisation and report on the performance gains/losses achieved. You are also required to explain why you believe the gain/loss in performance was observed.

Hint: Ensure that you have sufficient data in your database, or else you might not observe the gain/loss.

Task 8: Development ..... (25 marks)

Your project will be assessed, not only on the functionality it provides, but also on your overall development practices such as:

- Usage of git
- Data validation techniques
- Utilisation of a package manager
- Ease of understanding, spelling and grammar, and structure in your code, git commit messages, uploaded PDF, README etc.
- Quality of the overall delivered solution

Task 9: Demo ..... (40 marks)

Your team will be required to demo your solution to the lecturers of COS221. Booking slots will be opened closer to the time. Note, that all team members are required to be present for the demo to receive marks. In addition, each team member has to explain in the uploaded PDF all their contributions to the project. The demo will be evaluated based on functionality, user experience, presentation, and individual contributions.

Task 10: Bonus Task - Push the Boundaries ..... (15 marks)

This task is designed to encourage creativity, innovation, and independent learning. You may choose one or more of the options below, each of which introduces a new concept or advanced application beyond what was required in the main project. This is your opportunity to experiment, explore, and potentially impress with something unexpected. *The task and how it was implemented should be documented and explained in your submission.*

1. **Design-focused - Enhanced UI/UX**
   Create a polished, user-centered interface that:
   - Prioritises Accessibility, which could potentially focus on the following features:
     - Consideration for users with colour vision deficiencies
     - WCAG 2.1 compliance
     - Use of proper semantic HTML elements
     - Support for keyboard navigation and screen readers
     - any other features
   - Use of tools or methods to verify colour contrast ratios
2. **Security First – Secure Your App**
   Conduct a basic security audit and implement two or more improvements, such as:
   - Secure password hashing *(Be careful, common hashes exist and should elaborate your choice of hashing techniques and algorithms used).*
   - Login attempt rate-limiting

- CSRF/XSS protection
- Monitoring, through audit or error logs
- Session management

3. **Advanced Feature**
   Design and implement one unique feature not already required. Examples:
   - Gamified user interactions
   - Recommendation algorithm that is implemented to recommend products in the system
4. **Data Visualisation Dashboard**
   Create a basic dashboard to show insights, or audit trails such as:
   - user activity over time
   - graph visualisations for price history insights or another appropriate feature
5. **Advanced Git Features**
   Version control and automation are essential practices in modern software development. One way to achieve this is through Continuous Integration (CI), where automated testing plays a critical role. For this task, you will implement CI using GitHub Actions to streamline your development workflow and ensure code quality through automated testing.
   Note: This task will require a proper git branching strategy to work well.

IMPORTANT NOTES:

1. Please refer to the rubric for the detailed allocation of marks.
2. You will be required to demo your project.
3. If you do not demo, you will receive **0**.

9 Rubric for marking

Research	**20**
General overview and explanation	4
Different types or categories of Products and retailers explained	4
Information on how content is rated or categorised	4
Additional features/information provided (Recommendations, User Reviews)	4
References	4
(E)ER-diagram	**30**
Entities and Attributes	10
Complex and Derived Attributes	10
Relationships and Cardinality	10
Mapping	**30**
Regular Entity Types	3
Weak Entity Types	3
1:1 Relationships	3
1:N Relationships	3
M:N Relationships	3
Multivalued Attributes	3
N-ary Relationships	3
Specialisation and Generalisation	3
Unions	3
Correctness	3
Relational Schema	**30**
Visual Representation	10
Primary, Secondary and Foreign Keys	5
Constraints and Checks	5
Data Types and Length Constraints	5
Correctness	5
Web-Based Application	**55**
Functional Requirements	10
Media Management Features	10
User Account and Profile Management	10
Sorting and Filtering Options	10
Queries	10
SQL Injection Prevention	5
Sample Data	**20**
Explanation of Data Sources	5
Data Entry Methods (Script or Manual)	5
Quality and Relevance of Data	10
Analysis and Optimisation	**10**
Explanation	2
Optimisation	3
Interpretation	5
Development	**25**
Version Control (e.g. Git)	5
README Documentation	5
Overall Quality and Impression	15
Demo	**40**
Functionality and User Experience	20
Presentation Quality	20
Group cohesion	**10**
Total	**270**
Bonus Task	**15**