Ollscoil Mhá Nuad
Maynooth University

SEMESTER 2 (Coding Examination)
2023-2024

CS230
Web Information Processing

Dr. P. Healy, Prof. R.J. Farrell, Mr. M. P. McCormack

The total time allowed for the CS230 Coding Examination is **24 Hours**.

PLEASE NOTE THE FOLLOWING IN RELATION TO COPYING AND PLAGIARISM DURING OPEN BOOK EXAMINATIONS

This an Open Book Examination and all answers must be own work. Collaboration, and direct copying of online material is not permitted. All examination submissions will be subject to Maynooth University's Plagiarism Policy which is available at this URL: MU Plagiarism Policy. As part of the examination process you may be selected for interview to discuss your answers with the Examination Board. If you experience any connectivity issues that prevent you from submitting the exam on time you must inform the lecturer by email within 24 hours of the final submission time. Any emails past this time will not be considered. The email must detail clearly the nature of the issue. Where possible make copies of all work being included in the submission attempt.

Answer Descriptions: When you are requested to "describe" in a question, you must "describe in your own words, and using examples not selected from lecture material, and not directly copied from online sources". This means that you cannot copy textual explanations (from online or printed resources), modify them slightly, and paste as your own answer. This is plagiarism and will be reported to the examination board. Similarly, you must not copy code examples directly from resources and use as your answer. You must modify and use appropriately and contextually. Copying and reusing code in this fashion is plagiarism and will be reported to the examination board. When describing something, you should endeavour to provide clear, critical comments that demonstrate your understanding of the topic.

Code Inclusion: When you are asked to provide code examples they do not have to be fully syntactically or semantically correct. The code fragments should give a reasonable impression of how the solution might be constructed. You do not need to test code fragments included in an answer to ensure that they actually work; it is fine to have a few errors. The code fragments should have a sense of correctness, rather than being fully working code. Remember, code or protocol communication fragments are normally included to support your textual answer, and are not complete answers themselves.

PLEASE NOTE THE FOLLOWING IN RELATION TO THE MARKS BREAKDOWN FOR THE CS230 EXAMINATION COMPONENTS

For CS230, the marks split remains 50/50 for the Continual Assessment (CA) and Final Examination (FE). There will be a one-hour (max) in-person examination (for essay type answers on theoretical aspects of the module) which will contribute 20% of the final FE component, and there will be a one-day Coding Assignment which will contribute 30% of the final FE component. For the in-person examination, there will be two questions that require an essay form of answer, and will focus on theoretical aspects of the CS230 module. Each question will contribute a maximum of 10% to your final CS230 grade.

The Coding Assignment will involve creating a small database-driven web application incorporating a RESTful API (service). The Coding Assignment will contribute a maximum of 30% to your final CS230 grade. The marking breakdown for the coding test will be as

follows: 10% for a UI (User Interface - HTML/CSS/JS) Component, 10% for a Server-Side (RESTful API) Component, and 10% for a Database (DB) and DB-interaction Component.

PLEASE NOTE THE FOLLOWING IN RELATION TO THE TIMING AND DATES FOR THE CS230 EXAMINATION

The CS230 Final Examination Coding Assignment will be released on Moodle on Saturday 11th May 2024 at 09:00 and submissions will be accepted until Sunday 12th May 2024 at 09:00. You may only make one submission; there will be no exceptions to this rule on submissions. You will not be able to partially submit, continue to work and resubmit.

PLEASE NOTE THE FOLLOWING IN RELATION TO ISSUES OCCURRING DURING THE CS230 EXAMINATION CODING ASSIGNMENT

You will receive the paper or assignment via the Moodle Page for CS230. You can find this page using the “MY DASHBOARD” link on your Moodle landing page. Or you can click the link above. Moodle is the primary outlet for the examination. If you are worried about WiFi failure, download or take a screenshot of the Coding Examination assignment, so you have a copy to hand. There will be an examiner (Mark McCormack), available, via MS Teams messaging (“teams chat”), during the first two-hours duration of the coding examination, for two hours duration before the submission deadline, and for one-hour duration, after the deadline.

This messaging facility will be available for necessary exam related communication between students and examiners. It is not available for discussion about the actual Coding Assignment. Ensure you know how to use the messaging facility in MS Teams. MS Teams is the only technology that can be used for exam queries. Email will not be checked by the examiner during the examination.

CS230 - Web Information Processing Summer Examination Coding Assignment

This Exam Coding Assignment is worth 30% of Final CS230 Examination Mark. This is an open-book, graded examination coding assignment. You may use online resources for reference purposes only to help with the examination assignment. Please cite all references as comments in your submissions. You cannot directly reuse HTML/CSS/JS solution code from online sources. You must not engage with another student, in person or electronically (phone, social media, etc.) to secure assistance with this assignment. If you do so you will receive an automatic fail (0%). We will perform similarity checks on submitted assignments to check for collaborative efforts. A reasonable attempt at this assignment will gain you 30% of your final CS230 Examination Mark.

Coding Examination - Develop a Database, a REST API, and a User Interface (UI)

You are required to develop a solution for this coding assignment in three components:

(a) A solution database which appropriately models the data required for the coding assignment. The solution database may be implemented using either MySQL or MongoDB.

(b) A solution REST API that provides CRUD functionality for interacting with the solution database. The API may be implemented using PHP or NodeJS/Express.

(c) A User Interface (UI) which demonstrates (i) consumption of the REST API and (ii) implements the requested functionality detailed in the assignment brief that follows. You may implement your UI using PHP, a JavaScript or HTML framework (Plain HTML, React) or using ExpressJS in conjunction with a templating engine of your choosing.

Note that you do not have to use the same programming language throughout. It is perfectly acceptable to use PHP, say, to implement the UI and NodeJS/ExpressJS to implement the REST API, or vice versa. Or you may choose to develop an entirely PHP solution. It is important, however, that there is clear separation between the REST API and the UI that consumes the API.

Components (a), (b) and (c) are equally weighted at 10% for a fully functioning solution. You may decide to focus on one, two or three components. However, an important constraint for this assignment is that the components must be interoperable (work together) in order to secure maximum marks. Fully functional, standalone components will not receive 10%; they will receive 5% maximum, with the remaining 5% allocated to functioning correctly in a complete solution context.

Coding Examination - Requirements

You are required to develop a solution that implements a prototype Landlord-Tenant Management System for a hypothetical new set of apartments opening in Dublin. Your solution should provide a fully working UI (user Interface), REST API and DB (database) for the required functionality outlined below.

Your solution should support the creation and management of “Tenant” and “Landlord” personal information, together with support for the creation and management of “Tenant-Landlord Contract” information.

In terms of background, a “Tenant-Landlord Contract” (or “contract”) is the process of a tenant and landlord creating an agreement for the tenant to rent a given property with certain details. Landlords often own several properties, and your solution should provide functionality to record who they have made contracts with (one or more tenants), contract details and details about the tenants and landlords.

For Tenants and Landlords your solution should manage the following personal information:

1. Core Personal Details, for both Tenants and Landlords, including: Title* with options [Mx, Ms, Mr, Mrs, Miss, Dr or Other (specify)], First Name(s)*, Surname*, Phone Number*, Email Address* and Home Address* [Address Line 1*, Address Line 2, Town*, County/City*, EIRCODE]. When “Other (specify)” is selected, the form should provide a text-input field to record the user input for this detail. The fields marked * are required fields, i.e. they must contain values. You must validate forms prior to submission.

2. Additional Personal Details, for Landlords only, including: Date of Birth*, Permission to rent properties from the council* [Y/N] and permission for tenants to contact the landlord via email* [Y/N]. The fields marked * are required fields, i.e. they must contain values. You must validate forms prior to submission.

For Tenant-Landlord Contracts (contracts), your solution manages the following session information:

3. Contract Date*, Property Address*, Tenant(s)*, Landlord*, Fee (Monthly) (€)*, Property Door Number*, Contract Length* with options [Month, Year, Permanent] and Property Type* with options [Apartment, Semi-Detached, Detached, Other (specify)]. When “Other (specify)” is selected the form should provide a text-input field to record the user input for this detail. The fields marked * are required fields, i.e. they must contain values. You must validate forms prior to submission.

4. There will be at least one tenant in the contract, with a maximum of three tenants permissible. There will only be one landlord in the contract. Your solution is not required to provide a document upload facility.

5. As tenants normally renew their contracts, the contract date is important. Your solution is not required to automatically calculate the current date. This can be set manually using the online form.

Your solution's RESTful API (for example, created using PHP or NodeJS/ExpressJS/etc.) should provide CRUD functionality for Creating, Searching (Retrieving), Updating, and Deleting tenants, landlords, and contract information stored in either a MySQL or MongoDB database using online forms consuming the REST API.

You may implement your User Interface using HTML/CSS/JS, PHP, a JS Framework, or using ExpressJS in conjunction with a templating engine of your choosing. User Interface solutions may use jQuery and Bootstrap. Docker solutions are not permitted. Solutions realised using AngularJS or ReactJS, must provide detailed working (and tested) setup instructions with the submission, or it will not be possible to test; grading will then be based on a code review only.

Coding Examination - Additional Notes

(i) Please note that you should implement best practice when it comes to database design for this assignment, i.e., you may choose to have normalised or de-normalised models, or a combination approach depending on your choice of database management system (MySQL or MongoDB).

(ii) If you are using MySQL for storage your solution must use the Computer Science Department Webcourse service (https://webcourse.cs.nuim.ie). If you are using MongoDB for storage your solution must use the Atlas Cloud service (https://www.mongodb.com/atlas/database).

(iii) You may write, or reuse, previously personally developed functions to randomly create personal and address data; for speed it may be appropriate to use the NodeJS package, faker.js, as shown in Lecture 24 Lesson 02. Your solution should include complete personal data for, at least six therapists, at least ten clients, and at least twelve sessions.

(iv) Your code should include a brief description for the database design (your data modeling approach) and the impact on your REST API development. This should be included as a comment at the bottom of your REST API code submission.

(v) For this examination coding assignment your solution should validate data in both the User Interface and the REST API in order to secure maximum marks.

(vi) For this examination coding assignment you must generate online forms to collect and validate data sent to the database via a REST API. Please include an example of sample routes as a comment at the bottom of your UI code submission. Your solution should use styled tables, etc. to display retrieved data.

(vii)Your solution must “hard-code” your database authentication details into you application (in app or in a loaded file). If you are using the MongoDB Cloud Atlas database hosting service you must allow access from anywhere (whitelist 0.0.0.0). It will be necessary to access to your database in order to correct your assignment. Please note that you must use a remotely accessible database for this examination. If it is not possible to access your database you will not be awarded marks for this part of the coding assignment.

As a backup, please extract and include your database with your submission. Do not use a locally hosted database for this assignment. Please note that there are many sample (JS, PHP) solutions for implementing similar solutions, using REST functionality, available online. While it is fine to consult these, and accompanying articles, for references, you may not re-use code from these projects.

Please cite your reference sources in your codebase. We will search and identify online coding solutions to similar problems for the purposes of checking against submitted solutions in instances where we have concerns about code originality. It is fine to reuse

your earlier assignment code, but not the working code submitted previously by peers. We will consult earlier assignments if we have concerns about plagiarism.

IMPORTANT SUBMISSION DETAILS

Before submitting your coding assignment students should check that their solution works in Chrome and/or Firefox. Please indicate the Browser, Operating System (Linux/Windows/MacOS) and Browser version used for testing (as a comment in your submitted code). All work must be submitted via the upload link provided in Moodle (E:CS230[A]). Work submitted via other means will only be accepted by prior arrangement with Mark McCormack. All work MUST be submitted by the due-date deadline. Late submissions will not be accepted. The assignment submission is a document named “summer-exam-coding-assignment-zip” (containing your solution files together with any other resources used in the assignment solution. Please include a dump of the data from your database (as a text file) named “database.txt”. Please ensure that all external files use relative directory referencing, rather than hard-coding the files’ location. And ensure that you include your subdirectory files. Do not include the node-modules directory