Web Framework Development

Year 3 Computing

TU764 / TU860 / TU757

Project – Semester 2 2024/25

Individual Project

You are to design, develop, test, publish and defend a database-driven Django Python website, to illustrate the topics of the module. Your project should be an original approach to developing a website based on your case study for this project:

- NOTE: your case study must be confirmed with the lecturer.
- Projects for a case study that has not been agreed with the lecturer will not be graded.

You can create a new project from scratch or use an initial Django project following the guidelines on the official Django website.

Administrative

- Project submission deadline: 24/04/2025 11:59pm
- Should you need an extension in critical circumstances, please seek an approval from the lecturer before the project submission deadline. You might be granted an extension up to 48 hours. Late submission in this case might have penalties applied.
- Late submissions without approval are not allowed.
- The project results weigh 50% of overall module results.

DELIVERABLE 1: Github URL (no marks – but I can’t grade code without it!)

- Submission: URL of public/private Github project source code repository.
    - NOTE: if your Github project is private, you must add **Dr-AnhDuong-Trinh** as a collaborator to your private repository.
    - If your code cannot be accessed immediately after the deadline has passed, your work will be classed as late, and may not be accepted at all

DELIVERABLE 2: Project Design (10% of the MODULE grade)

- Submission:
    - PDF file in Github repository Design1_UseCases.pdf – a Use Case diagram, for the use cases that you have implemented in your submitted

project
  - PDF file in Github repository Design2_ERD.pdf – an Entity-Relationship Diagram, for the /src/Entity classes implemented in your submitted project
  - Structured text describing case study use case tests Design3_tests.zip – BDD Cucumber User Stories, Features & Scenarios (tests for each feature in separate text file)

DELIVERABLE 3: Project Implementation (40% of the MODULE grade)

- Submission: All the code and tests of your project in the Github repository
  - If you have published your project then state the URL of your published website in the Github REAME.md document this must be up and publicly available for the day of the deadline and all the following 48 hours (this is possible using a free trial app on Fortrabbit)
  - The published website should exactly match the contents of your final Github commit, with content from a working database and logins etc.

GRADING

All students are allocated date/time and venue to showcase their project for grading purposes.

- NOTE: If you do not attend a grading/defence session your project will not be graded and score zero.

Required fixtures

- You must include database setup fixtures for EVERY Entity classes
- Ensure you create fixtures that relate objects of different entities
- Please provide simple fixtures for your Users as follows:
  - Make all users have password of “password”
  - ROLE_ADMIN = user name admin, or admin@amin.com for email
  - ROLE_USER = user name user1, or user1@user1.com for email
  - And so on …

General criteria for web project grading

- **Originality** (your own work)

- **Correctness** (it works)
  - Your project should be error free

- **Completeness** (features specified are present)
  - Refer to the marking grid
  - The project submitted should offer use cases for non-trivial actions users in different roles can perform

- **Code Quality & Software Engineering Process**
  - Good quality and meaningful identifiers (names) files / folders / classes / ids
  - Follow correct UpperCamelCase / lowerCamelCase for OOP in Python
  - Decent and consistent indentation and code layout
  - Demonstrate automated TESTING

- **Usability**
  - A decent user experience: e.g. use Bootstrap CSS
  - It should be straightforward and easy for users to locate and perform each use case

- **Consistency & Coherence** – decent website 'look and feel' (please avoid drop-down nav-bars)
  - A simple but effective website will score better than a hard to understand / messy complicated website – so keep a non-technical website visitor in mind at all times.

- **Technical Challenge** – Demonstration of attempting interesting / challenging features
  - Note: there is no extra credit for using JavaScript, so don't make work for yourself

- **Value Added** (more than just examples reproduced or on changed in minor ways)

Project Requirements

You are assigned your project theme and topic based on your student number. Should you request the change of theme and topic, you must discuss it with the lecturer and get approved by them.

The project themes and topics are retrieved from the list of industries and business areas as follow:

	Theme		Must-have Use Case
0	Agriculture	0	Customer Support (Customers – Cases)
1	Sports	1	Invoicing (Invoice – Items)
2	Health	2	Purchase Orders (PO – Items)
3	Advertising	3	Logistics (Parcel – Contents)
4	Leisure	4	Inventory (Item – Categories)
5	Retail	5	Payroll (Employee – Entries)
6	Finance	6	Recruiting (Candidate – Skills)
7	Airlines	7	Training (Module – Students)
8	Food	8	Sales (Orders – Items)
9	Education	9	Productivity (Projects – Tasks)

Topic assignment

- Theme: Using the second last digit of your student id, please select a topic from the Theme column.
- Use case: Using the last digit of your student id, please select a use case from the Must-have Use Case column.

Example: If your student ID is Bxx45, you must create a web project in Leisure theme with the use case Payroll (Employee – Entries) among your use cases.

List of project requirements

- Your web framework should have at least 3 non-Admin/User roles
- Your web framework should have a good range of use cases, e.g. at least 5 use cases, that include your must-have use case from above
- Your project contains a good database design
- Your framework has automated testing functionalities
- Your progress is tracked on a regular basis, e.g. weekly commits to your Github
- And so on (please refer to the marking grid for reference)

Academic Honesty

By submitting your project for assessment, you agree to the following:
"The material contained in this assignment is my own original work, except where work is clearly identified and duly acknowledged. No aspect of this assignment has been previously submitted for assessment in any other unit or course."

In general: For every piece of work you submit to the Institute, your documentation must make it very clear which parts are your own creations; the work of others; and your adaptation of other's work, and what your adaptations were. Work submitted without full and unambiguous acknowledgements is plagiarism. Plagiarism and academic dishonesty can lead to failure of the module and other penalties outlined in the Institute's rules and regulations. For any project or coursework, you should discuss how to best declare the use of work from other sources with your lecturers.

INDIVIDUAL PROJECT: The work you submit must be your own (with fully declared exceptions described below). It is fine for you to ask a lecturer or fellow student for assistance with some problem you are stuck on during the project, but the actual final work created and submitted must be your own. While you may get IDEAS for code to solve particular problems from other sources, you must write and user the command line tools to generate all codes yourself. Remember by submitting this project you are declaring that it is all your own individual work unless explicitly and unambiguously acknowledged by you.

- You must NOT SHARE YOUR OWN CODE FILES with fellow students! Also ensure your project Github repository is private!
- You should NOT BE TYPING IN CODE into a fellow student's computer project – talk to them yes, but let THEM design and type-in their own code.

You may NOT use other Python code inserted into your own classes or scripts - i.e. all your Python classes must all be your own code, or generated by you at the command line, or imported through the Composer package manager utility.

The text content of your web pages should all be prose text that YOU have authored yourself (so do NOT copy and paste from Wikipedia or IMDB etc.) – pages only need a few sentences/paragraphs each, so write and paraphrase them yourself in your own words. Or you may generate random content, e.g. with Faker.

You may use the following without need of citation:

- You may use and modify any example codes from the lectures/labs
- You may generate code with the command line tools
- You may (and in fact are encouraged to) use full Python and Django components which you install

You may use the following, with citation in a /public/sources.txt document

- Client-side media files, such as images, fonts, and templates for HTML/CSS/JavaScript. Give the URL and list what client-side files you used from that location

Talk to the lecturer if you have any doubt about whether it is permitted to use / how to properly cite resources you have not created yourself for this module project.

Declaration of media sources

/public/sources.txt

/public/sources.txt – text file declaring sources

- You are to submit a text file which states the origin of each image (and font) used in your website (be specific, e.g. saying “Google images” is not good enough! You need to state the URL including the original filename of the image from that source, and also state the image filename as it appears in your website folder)
- If you created an image yourself, say so.
- You should also declare any other sources used in your website, and state clearly how they have been used, so it is clear which parts of your project are your own work, which are the work of others, and the extent to which you have changed any work from others
  - But remember, apart from images all other parts of the project should be your own original work.

Example of content

- File: /public/sources.txt

```
logo.gif
http://www.itb.ie/images/itb_logo.gif

cat.jpg
http://images.wisegeek.com/young-calico-cat.jpg
```

- And so on
- All you need is to state the name of the image file in your images folder, and the URL of where I can find the original image on the web image
- HINT: View Source will give you direct links to images that you see on a web page. Click that link and you should see the image, and be able to copy and paste the URL into your sources.txt document.

NOTE

- Do NOT submit a PDF or Word document!
  - Just a TEXT FILE with simple contents as shown above
- When you download an image / CSS file, that is a good time to rename it as you save it into your \web\images directory
  - You’ll be marked down for poorly named images like: p041gljk.jpg from: http://ichef.bbci.co.uk/wwfeatures/wm/live/1280_640/images/live/p0/41/gl/p041gljk.jpg
  - A much better file name would be: cat_face_closeup.jpg from: http://ichef.bbci.co.uk/wwfeatures/wm/live/1280_640/images/live/p0/41/gl/p041gljk.jpg