* In this assignment, you will use Django to build a highly *interactive* website that addresses the requirements outlined in the **Project Brief** document.  
    
  The website will utilize a database to record and retrieve information. You may consider this a significant enhancement to your previous Django assessment submission.  
    
  The website must provide full C.R.U.D. functionalities (Create, Read, Update and Delete) for the records within your database. In other words, you will create your version of the Django Admin App which deals exclusively with the maintenance of your data.  
    
  More details about your tasks are given in the following section.
* **Tasks  
    
  1.** Address **Project** **Objective 2** (see the **Project Brief** document) and ensure that your website meets the following criteria:  
  + Be created as a Django version 4.0 project (or any later version), containing at least one Django app.
  + A homepage that includes:
  + The names and student numbers of all group members.
  + A paragraph that promotes the key features of the website.
  + Navigation options that provide easy access to the website’s home page, and other logical or helpful URL’s within your project. The navigation options should be available at all times throughout your website.
  + Use Django to store your information dynamically in a local database.
  + Use Django ModelForms on your website to accept the user input which will affect the database.
  + Full CRUD functionality is available on your website for each table (entity) in your data model.
  + All hyperlinks must be constructed with the use of named URL Paths and {% url %} template tag. Any hyperlink which points to resources outside of your website project is exempt from this requirement.
  + The website should use standard HTML5 syntax and essential HTML formattings, such as headers, tables, hyperlinks and text formatting where appropriate.
  + The minimum data complexity requirement must be met (a minimum of 3 related classes/tables in your data model).
  + Data model classes must include the supporting structure for the relationships between the classes. For example, models.ForeignKey( … ) and/or models.ManyToManyField(… ).
  + A sample data set must be supplied with the project submission containing a **minimum of 5 primary records and a** **minimum of 4 secondary records associated with each primary record**.
* **2.** Write a report (not more than 1,000 words) that reflects on the quality of your Django website, the effectiveness of your software development process, and the current team dynamics. Use this opportunity to revisit the Contract and the first Report as a team, including assessing the extent to which the previously suggested improvements for the final assessment have been achieved. Your instructor may also look for evidence in terms of personal, technical, and professional growth.
* **Grading**Unless stated otherwise in your Group Contract, all members equally share the grade for this assignment. Your instructor reserves the right to differentiate individual grades where appropriate.  
    
  The design of this assignment's marking rubric incorporates elements of existing rubrics used in the University of Edinburgh.
* **GitHub Classroom**GitHub Classroom must be used to manage Python coding collaboration for this assignment. Individual coding contributions must be visible on the GitHub Classroom. **This is compulsory.**
* **Submissions**Submit your work via Learnline **by the latest 23.59 ACST** on the stated due date of this assignment.  
  + Asingle **.zip file** containing your Django project that addresses Task 1. **Exclude** the virtual environment folder from this file.
* + A single Microsoft Word document (.docx) of no more than 1,000 words that addresses Task 2.
* **Email submissions will not be accepted.**
* **Unacceptable elements of submissions**You **must not** reuse the codes from our class exercises and lecture slides. They are your instructor's codes, not yours. Your instructor can quickly tell his codes from others. Reusing your instructor's codes may constitute an academic breach.  
    
  You **must not** obtain, re-use, or modify existing Web templates (including HTML templates).  
     
  You **must not** use any other Web framework than what is taught in this unit.
* **Submissions that do not run**It is **the student's responsibility** to ensure that their submissions run on the instructor's computer as intended, including cases where additional Python packages need to be imported. Submissions that fail to run **will be severely marked down**.  
    
  It is recommended that you simulate a submission with your peers before making the final submission.  
  + Make sure you are using the correct Django version
  + Make sure you have included all required files and Python packages
  + Make sure your .zip file is valid *before* you upload it to Learnline