HIT237 Assignment 2

Due On: Friday, Week 11.

Task 1 - Build an interactive database driven website

For this task you must create a website that uses Django forms in conjunction with a database to record information about a topic of your choice. You may select any topic of your choosing (with some exceptions) so long as it meets the complexity requirement outlined below.

Your website must provide full CRUD (Create, Read, Update and Delete) functionality for the records within your database. In essence, you will be creating your own small version of the Django Admin App which deals exclusively with the maintenance of your data.

Minimum Data Complexity Requirement

When choosing your topic it must have a complexity that allows you to have a minimum of 2 **additional** classes (tables in the database) associated with the principal topic. These classes must have an identifiable relationship with each other. For example, if you chose *Hair Care Products* as your topic, you could have the Product (our topic), Manufactures (which create this type of product) and Retailers (which stock this type of product). If you chose *Cooking Recipes*, you could have the Recipe (our topic), Ingredients (which are present in the recipe) and Required Tools (used to prepare the recipe). Use your imagination and chose a topic that you have genuine interest in.

Choose a new topic for your website that is different from the topic you chose in Assignment 1. Do not use the above examples, or other examples given during lectures, as your topic. If you are having trouble choosing a topic, please speak with your lecturer for assistance.

Task 1 Requirements

Your project must meet the following criteria:

- Be created as a Django v1.11.2 project, with at least one App (most will only need one app).
- A landing page (sometimes called a home page) that includes:
  - Your name.
  - Your student number.
  - A brief paragraph (a few sentences) that talks about your principal topic, and how the associated tables are important in relation to that topic.
- 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 Forms on your website to accept the user input which will affect the database.
- Full CRUD functionality available on your website for each table in your data model.
- All hyperlinks must be constructed with the use of named URL Paths and {%url%} tag. Any hyperlink which points to resources outside of your own website project is exempt from this requirement.
- The website should use standard HTML5 syntax and make use of normal HTML formatting, such as headers, tables, hyperlinks and text formatting where appropriate.

- 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. Eg, models.ForeignKey(…) and/or models.ManyToManyField(…).
- A sample data set must be supplied with the project submission containing a **minimum of 6 records** for the principal topic chosen, along with any required secondary records. The number of secondary records will depend on your data model and chosen topic.

Task 2 – Documentation

Create a PDF file containing the following elements which document your project:

- A high level ERD (Entity Relationship Diagram) showing the cardinality between the classes (tables) in your data model. This should be a quick diagram similar to the example from slide 3 of the week 7 lecture.
- A detailed data dictionary for each class in your data model. Refer to the lecture from week 5 for minimum expected information to be included.
- A brief user guide on how to use your website to Add, Update & Delete a record. The guide only needs to be for 1 of the classes/tables in your project. You can include small images to provide clarity if you wish (not a requirement). Using the *Cooking Recipe* example from Task 1, I could create a user guide on how to Add, Update & Delete records from my Ingredients table.

Submission

All submissions should be completed via Learnline before 11:59pm on the due date.

Please Note: Due to administrative requirements prior to exams, the due date for this assignment is **fixed** and will not be altered. Any request for extension will be required to show suitable cause in accordance with University Policy.

Assignments That Do Not Run

If for any reason I am unable to run your assignment, it **will not be marked** and you will receive a zero grade for the assignment.

For Assignment 1 we were more lenient and tried to get your projects running in order to mark them for you. However, there are extreme time constraints on the grading of Assignment 2 and you have had ample opportunity to learn what is required to make a Django project run.

Make sure you are using the correct Django version, make sure you have included all required files and make sure your ZIP file is valid *before* you upload it to Learnline.