



ISCG7420
Web Application Development
Assignment 1
Semester 1, 2020

**School of Computing, Electrical and
Applied Technology**

Due Date: Sept 27th 2020 8:00PM

Total Marks: 100
Course Weighting: 60%

Learning outcomes covered in this assignment

1. Discuss the philosophy of client-server computing and its impact to the computing industry.
3. Design and implement a dynamic web application using a range of languages/technologies/tools.
5. Design and develop a database client-server solution that meets specified organisational requirements using database and modern data access technologies.

Assignment instructions:

- You will work individually on this assignment.
- Select an idea from appendix 1. For the selected idea you will develop a Django web application. If you have your own idea discuss it with your lecturer and get their approval.
- You will identify the required database models for the given idea, then create a dynamic web application using a modern web application framework.

Assignment Submission:

- Use the *Assignment 1* directory of your class GitHub repository to store your assignment and add your lecturer as a collaborator. Periodically throughout the course your lecturer will clone your repository and check your progress. On the assignment due date (and for up to 72 hours afterwards) your lecturer will download the most recent version of your assignment. You are required to ensure that GitHub always has the latest version of your code.
- If you forget to push your most recent changes by the due date, then the lecturer will use the version available on GitHub when the deadline is reached for marking. **Make sure you regularly submit files. Submitting an assignment with no in-progress code is highly suspicious and will be investigated. If you use and submit code that you didn't write, you will fail this assignment, and fail the course. Ask your lecturer for help long before the due date instead of plagiarizing.**

Instruction:

For your selected idea you need to design and implement a dynamic web application using Django, PostgreSQL, Redis and other available web development tools. You will also host your web application on Heroku with available add-ons. You will upload your code to GitHub.

Task 1: Setup – Due Sun Aug 30th**[15 Marks]**

You need to do the following setup before creating your web application.

Create a repository on GitHub for your web application. [1 mark]

Add a detailed **README** describing in at least three paragraphs what your web application will do or what problem it solves. [2 marks]
Use additional diagrams where appropriate.

Create a minimum of **10 descriptive tasks** on GitHub that break down what you will need to do, and add rough time estimates to each task. Estimates should be no longer than 8 hours, so break them into smaller tasks. [2 marks]

Discuss in a paragraph in your README what is **server-side rendering** and how does it work. [1 mark]

Create a **developer journal with 30 days of entries** in a JOURNAL.md file in your GitHub repository. Do a little bit of work every single day and record everything you do for the tasks. Store it in a markdown in your assignment GitHub repository. See appendix 2 for a sample developer journal. [5 marks]

Create database **Entity-Relationship Diagrams** (ERD, classes and relations) for your web application models and embed the images in your README. [4 marks]

Task 2: Django Framework Part 1 – Due Sun Sep 13th**[25 Marks]**

Your web application must include the following:

Create a minimum of **10 models**, which must demonstrate use of foreign keys (many-to-one), one-to-one and many-to-many relationships. [5 marks]

Create a minimum of **10 templates** and demonstrate an understanding of **extending a base template, including other templates, for-loops and if/else** in some of your templates. [5 marks]

Create a minimum of **10 database migrations** for your models. [3 marks]

Create and use a minimum of **10 views**. [5 marks]

Create **4 forms** (using both ModelForms and Formsets) and embed them in your templates. [4 marks]

Create user **signup, login/change password, forgot password**. [3 marks]

Task 3: Django Framework Part 2 – Due Sun Sep 20th**[25 Marks]**

Demonstrate use of the ORM features and **queriesets**. You must use **filter, exclude, order_by** and related field lookups [5 marks]

Use **select_related** to optimize at least one query. [1 marks]

Use of both **function based** and **class based** views. [4 marks]

Customise the administrator control panel. You must demonstrate use of **list_display, search_fields, fields, list_filter, custom_fields** and **inlines** or custom forms where applicable. [5 marks]

Demonstrate use of Django **messages** [2 marks]

For each model, export a corresponding **.json** file with the **dumpdata** command. [1 mark]

Create **10 unit tests** and **use test fixtures** from the previous command. [5 marks]

Use Django **signals** to print a message to console when a user is created. [2 marks]

Task 4: Deployment – Due Sun Sep 20th

[10 Marks]

Deploy your web application on Heroku.

[6 marks]

Create a **PostgreSQL instance** on Heroku and connect it your web application.

[4 marks]

Task 5: Additional features – Due Sun Sep 27th

[25 Marks]

Add the following additional features to your application:

Setup and configure **redis** on Heroku as an application cache

[3 marks]

Setup and use **celery + redis** as a task queue for your application.

[5 marks]

Setup and configure Amazon S3 to allow images / media to be uploaded from within your application.

[5 marks]

Setup Sendgrid to send an **email** to users upon **signup**.

[3 marks]

Setup and use Sentry to track Django exceptions/errors.

[3 marks]

Setup a custom domain name and configure DNS for your application, use GitHub Education pack for a free one, and choose a domain that you can reuse after this course is completed.

[4 marks]

Setup and use Cloudflare. Ensure that your DNS record points to CloudFlare and not Heroku.

[2 marks]

Marking Schedule

Task	Marking Criteria	Marks	Given	Comments
1	<p>GitHub Repository is created.</p> <p>Detailed README file with description on what web application will do or what problem it will solve (minimum 3 paragraphs). Additional diagrams provided where appropriate</p> <p>At least 10 descriptive tasks are created on GitHub. It includes</p> <ul style="list-style-type: none"> - break down of what need to be done - rough time estimates of each task <p>Working of server side rendering for web application is discussed</p> <p>Developer journal</p> <ul style="list-style-type: none"> - Has 30 days worth of entries - Mentions problems encountered and solutions used - References online resources used to solve problems - Has code snippets of errors or things learned during the project <p>Create database ERD (classes and relationships) for your application models</p>	<p>1</p> <p>2</p> <p>2</p> <p>1</p> <p>5</p> <p>4</p>		
2	Minimum 10 models using foreign key, one to one and many to many relationships among models.	5		

	Minimum 10 templates are created	5		
	Minimum 10 data migrations created	3		
	Minimum 10 views are created and used	5		
	4 forms are created and use of ModelForms/Formsets is demonstrated	4		
	User signup/login/change password/forgot password	3		
3	Use of the ORM features and querysets are demonstrated. It includes use filter, exclude, order_by and related fields	5		
	select_related is used to preload data for at least one query.	1		
	Uses both function-based and generic class-based views	4		
	Administrator control panel is customised. It must display <ul style="list-style-type: none"> • list_display • search_fields • fields • list_filter • custom_fields • inlines or custom forms where applicable 	5		
	Use of Django messages demonstrated	2		

	Dumpdata command is used to export data to JSON files	1		
	Unit/integration tests are created and test fixtures used	5		
	Use of Django signals demonstrated	2		
4	Web application is deployed on Heroku	6		
	PostgreSQL on Heroku is created and connect web application	4		
5	Redis on Heroku is used as an application cache	3		
	Celery + redis is setup & configured for task queue	5		
	Images / media uploaded to Amazon S3	5		
	Sendgrid & Django configured to email users on signup	3		
	Django errors are tracked using Sentry	3		
	Custom domain name is setup and DNS is configured for the application	4		
	Cloudflare is setup and used for protection	2		
Total Marks		100		

Late Submission of Assignments:

Assignments submitted after the due date and time without having received an extension through Affected Performance Consideration (APC) will be penalised according to the following:

- 10% of marks deducted if submitted within 24hrs of the deadline,
- 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline,
- 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline,
- No grade will be given for an assignment that is submitted more than 72hrs after the deadline.

Assistance to other Students:

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are unacceptable in an assessment.

Beneficial Assistance:

- Study Groups
- Discussion
- Sharing Reading Material
- Reading the available online and library resources

Unacceptable Assistance:

- Working together on one copy of the assessment and submitting it as own work
- Giving another student your work
- Copying someone else's work, this includes work done by someone not on the course
- Changing or correcting another student's work
- Copying from books, the Internet etc. and submitting it as own work; anything taken directly from another source must be acknowledged correctly; show the source alongside the quotation
- Don't copy code from a website or video tutorial and pretend you made it or slightly change it. This will be an instant fail (0%).

Appendix 1

Sample Project Ideas (Simple versions):

1. Instagram clone
2. Facebook clone
3. Twitter clone
4. YouTube clone
5. Pinterest clone
6. Reddit clone

You must identify and construct the models required by most modern web applications. Example functionality is for example: making posts, uploading images, liking posts, comments, sending messages, adding a tag to a post, notifications, login/logout, etc.

If you choose to do something different you need to get approval from your lecturer. Communicate with your lecturer during the course to ensure your application meets the required guidelines and marking schedule.

Appendix 2

Developer Journal Example:

===

March 23rd 2020 – Started @ 2:30PM

Going to work on Task #5, to try and implement saving user data to text files.

Estimate: 30 minutes

Got a bug, file isn't being created correctly.

5:15PM Still stuck on bug. Switching to Task #7 to add a button.

5:45PM Added a button, closed Task #7.

7PM: Fixed bug with file not being created. I was accidentally saving it to the wrong directory.

Future Suggestion: Make sure to always check which directory I save files to.

Actual time taken: ~4 hours.

7:30PM: Need to create an Invoice class which has save() method.

Estimate: 1hr

Stopped @ 9pm.

===

March 24th 2020 – 10AM

Continuing on Task #6 from yesterday – Creating class for Invoice

10:30AM: Finished class for Invoice (Task #6).

Actual time taken: ~2 hours.

10:45AM: Debug Issue #12 – Game collisions aren't working and object disappears off screen.

Estimate: 4 hours

1PM: While debugging game collisions I realized that I forgot to enable logging. I enabled logging and instantly saw why collisions weren't working. I wasn't checking if a collision occurred before moving the object.

Future Suggestion: When creating classes, start with a list of pseudo-code comments of things I need to remember to check. Also don't forget to enable logging when debugging.

Actual time taken: ~2 hours

2PM: Refactored / rewrote the do_stuff() function to be smaller and more clear.

4PM: Found a StackOverflow post for a possible solution to Task #9:

<https://stackoverflow.com/questions/1077347/hello-world-in-python>

5PM: Solution didn't work. Out of time for this feature so will cut it and work on a different task.

Etc.. Write in your journal and work on your code daily.