Unitec
Institute of Technology
TE WHARE WĀNANGA O WAIRAKA

School of Computing, Electrical and Applied Technology

ISCG7420
Web Application Development

Assignment 1

Semester 1, 2020

Due Date: June 1st 2020 11:59 PM

Total Marks: 140
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:

- Create a GitHub repository for 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.

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 [24 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 minimum 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 how server-side rendering works for a web application. [3 marks]

Create a developer journal 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. [10 marks]

Create Entity-Relationship Diagrams (ERD) for your web application models and embed the images in your README. [6 marks]

Task 2: Django Framework [97 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. [10 marks]

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

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

Create and use a minimum of 10 views [10 marks]

Create 5 forms and demonstrate use of ModelForms/Formsets [5 marks]

Create user signup, login/change password, forgot password, Emails sent on signup (Sendgrid). [5 marks]

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

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

Demonstrate use of both function based and class based views [2 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. [10 marks]

Setup/configure S3 to allow images / media to be uploaded from within your application. [10 marks]

Demonstrate use of Django messages [2 marks]

Setup and use celery as a task queue for your application [5 marks]

Create unit/integration tests and use test fixtures [10 marks]

Demonstrate use of Django signals [2 marks]

Task 3: Deployment [7 Marks]

Deploy your web application on Heroku. [3 marks]

Create a PostgreSQL instance on Heroku and connect it your web application. [3 marks]

Export your data to JSON files using the dumpdata command. Ensure they're stored in your GitHub repository, and make sure you only export relevant tables. [1 marks]

Task 4: Additional features [12 Marks]

Add the following additional features to your application. To do this you must:

Setup and configure Redis on Heroku as an application cache [3 marks]

Setup and use Sentry to track Django exceptions/errors [3 marks]

Setup a custom domain name and configure DNS for your application [5 marks]

Setup and use Cloudflare for protection [1 marks]

Marking Schedule

Task	Marking Criteria	Marks	Given	Comments
1	GitHub Repository is created.	1		
	Detailed README file with description on what web application will do or what problem it will solve (minimum 3 paragraph). Additional diagrams are provided where appropriate	2		
	At least 10 descriptive tasks are created on GitHub. It includes - break down of what need to be done - rough time estimates of each task	2		
	Working of server side rendering for web application is discussed	3		
	Developer journal - Has at least 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	10		
	Create ERD for your application models	6		
2	Minimum 10 models using foreign key, one to one and many to many relationships	10		

	among models.			
	Minimum 10 templates are created	10		
	Minimum 10 data migrations created	10		
	Minimum 10 views are created and used	10		
	5 forms are created and use of ModelForms/Formsets is demonstrated	5		
	User signup/login/change password/forgot password/ Emails sent on signup (Sendgrid) created	5		
	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		
	Use of function and class based views are demonstrated	2		
	Administrator control panel is customised. It must display • list_display • search_fields • fields • list_filter • custom_fields • inlines or custom forms where applicable	10		

	Images / media uploaded to Amazon S3 Use of Django messages demonstrated Celery to queue Message is used Unit/integration tests are created and test fixtures used Use of Django signals demonstrated	10 2 5 10 2		
3	Web application is deployed on Heroku PostgreSQL on Heroku is created and connect web application Dumpdata command is used to export data to JSON files	3 3 1		
4	Redis on Heroku is used as an application cache Django errors are tracked using Sentry Custom domain name is setup and DNS is configured for the application Cloudflare is setup and used for protection	3 3 5 1		
Total Marks		**140**		

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:

1. Simple Instagram clone
2. Simple Facebook clone
3. Simple Twitter clone
4. Simple YouTube 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.