

Level 5 Web Application Development: Unit 3 - Back End Development – Grading Sheet

Information

Name	Manal Abdulqawi		
Due Date		Date Submitted	
Grade Achieved	Distinction	GitHub Repo	https://github.com/ManalAbdulqawi/artwork-online-shop?tab=readme-ov-file

Criteria Breakdown

Criteria Code	Criteria	Achieved	Comments
1.1	Design a Full Stack web application to be built using the Django framework and to incorporate a relational database and multiple apps (an app for each potentially reusable component).	Yes	A full stack web app has been designed to a very good level of detail with wireframes and ERD included
1.2	Design a front end for a Full Stack web application that meets accessibility guidelines, follows the principles of UX design, meets its given purpose and provide a set of user interactions.	Yes	Some frontend components have been designed in the form of wireframes and are of acceptable quality

1.3	Develop and implement a Full Stack web application built using the Django framework, to incorporate a relational database, an interactive frontend and multiple apps (an app for each potentially reusable component).	Yes	The application has been developed and implemented as per the designs
1.4	Implement at least one form, with validation, that allows users to create and edit models in the Back end.	Yes	Multiple forms have been created which are reflected in the front end
1.5	Build a Django file structure that is consistent and logical, following the Django conventions.	Yes	Django file structure is consistent and well thought out
1.6	Write code that clearly demonstrates characteristics of 'clean code'.	Yes	Code is clear, concise and easy to understand
1.7	Define application URLs in a consistent manner.	Yes	Application URLs have been defined in a consistent manner
1.8	Incorporate a main navigation menu and structured layout.	Yes	Website has a structured layout and good navigation
1.9	Include custom logic that shows proficiency in the Python language.	Yes	Custom logic has been implemented
1.10	Write Python code that includes functions with compound statements such as if conditions and/or loops.	Yes	Compound statements used to implement custom logic
1.11	Design and implement manual or automated test procedures to assess functionality, usability, responsiveness and data management within the full web application.	Yes	Manual testing has been performed to ensure that the web app functions as expected
2.1	Design a relational database schema with clear relationships between entities.	Yes	Designs have been created for the database with clear relations between the entities
2.2	Create at least TWO original custom Django models.	Yes	Multiple custom models have been created

2.3	Create at least one form with validation that will allow users to create records in the database (in addition to the authentication mechanism).	Yes	A number of forms have been created all of which include validation
2.4	Implement all CRUD (create, select/read, update, delete) functionality	Yes	CRUD actions have been implemented across the website where relevant
3.1	Implement an authentication mechanism, allowing a user to register and log in, stipulating a clear reason as to why the users would need to do so.	Yes	Users are able to login and register
3.2	Implement log-in and registration pages that are only available to anonymous users.	Yes	Login and register pages are not viewable to anon users
3.3	Implement functionality that prevents non-admin users from accessing the data store directly without going through the code.	Yes	Non-Admin users cannot access data they shouldn't be able to
4.1	Implement at least one Django app containing some ecommerce functionality using an online payment processing system (e.g. Stripe). This may be a shopping cart checkout, subscription-based payments or single payments, donations, etc.	Yes	Stripe has been implemented as part of the application to process one-time payments
4.2	Implement a feedback system that reports successful and unsuccessful purchases to the user, with a helpful message.	Yes	Toast alerts implemented to provide the user feedback
5.1	Deploy the final version of your code to a hosting platform and test that it matches the development version.	Yes	Application has been deployed through Heroku
5.2	Ensure that all final deployed code is free of commented out code and has no broken internal links.	Yes	No commented-out code or broken internal links present
5.3	Ensure the security of the deployed version, making sure to not include any passwords in the git repository, that all secret keys are hidden in	Yes	Git repo has been kept secure

	environment variables or in files that are in gitignore, and that DEBUG mode is turned off.			
5.4	Use a git-based version control system for the full application, generating documentation through regular commits and in the project README.	Yes	Version control used throughout with a project readme created	
5.5	Create a project README that is well-structured and written using a consistent markdown format.	Yes	Readme is clear and well formatted	
5.6	Document the full deployment procedure, including the database, and the testing procedure, in a README file that also explains the application's purpose and the value that it provides to its users.	Yes	Full development process has been documented	
Merit 1	<p>Design and build a realworld Full Stack MVC application with a Front end:</p> <ul style="list-style-type: none"> • that is easy to navigate and allows the user to find information and resources intuitively. • where the user has full control of their interaction with the application. • where all data (CRUD) actions are immediately reflected in the user interface. • where the purpose is immediately evident to a new user. • which provides a good solution to the user's demands and expectations. • which has a clear, welldefined purpose addressing the needs of a particular target audience (or multiple related audiences). 	Yes	Application has been deisgn and developed in a way that fits the required as defined within this criteria. App is of a good quality.	
Merit 2	Produce a robust codebase	Yes	Code is robust and can handle cases well	

Merit 3	Follow a Test Driven Development (TDD) approach (for JavaScript and/or Python), demonstrated in the git commits	Yes	It is evident where you have made changes to fix errors from the commit messages such as correcting the responsiveness for smaller screens
Merit 4	Configure the project efficiently through wellkept Profile, requirements.txt file, settings files, keeping the data store configuration in a single location where it can be changed easily.	Yes	All the required files are present and correctly made
Merit 5	Fully describe the data schema in the project README file.	Yes	Database schema has been well defined within the readme
Merit 6	Demonstrate solid understanding of Django template syntax, logic and usage, placing Django logic in the component where it is best suited (e.g., data handling logic is in the models, business logic is in the views).	Yes	The code written displays a solid understanding of Django and implements a complex and featureful website
Merit 7	Use version control software effectively to provide a record of the development process.	Yes	GitHub has been used throughout to properly record the development process
Comments Towards Distinction	A very good submission with a featureful website developed. You've implemented tags and other subcategories and the user is also able to effectively search for products and sort results. Code is of a very good quality with a good layout and is easy for a 3 rd party to debug. Well done!		