10.0 hours
Project Status Ready for Submission

Project 4

Store Inventory

Are you ready to have your project reviewed? Before you submit your project, make sure you have double-checked your work. We want you to pass the first time!

If you get your project back and it needs work, you'll have to wait 1 day before you can resubmit your project.

[Submit for Review](#)

- [Instructions](#)
- [How you'll be graded](#)

To get a passing grade you'll need to submit a project that meets the following requirements. If your project receives a "Needs Work" rating on any of these tasks, then you'll need to edit your project and re-submit it until it "Meets" or "Exceeds" each task requirement.

	Needs Work	Meets Expectations	Exceeds Expectations
CSV to Database	One or more of the items below were not cleaned or inserted as the proper type. • `product_quantity` must be an Integer. • `product_price` must be an Integer. • `date_updated` must be a Date object.	All data from `inventory.csv` was read and cleaned before adding it to the database. • `product_quantity` was stored as an Integer. • `product_price` was stored as an Integer converted to cents *(Ex. $3.19 becomes 319)* • `date_updated` will be stored as a Date object.	N/A

	Needs Work	Meets Expectations	Exceeds Expectations
Database setup	One or more of the following items were missed: • The Sqlite database was not named `inventory.db`. • The products table was not created in the database using the `Product` model defined	• A Sqlite database named `inventory.db` was created • The products table was created in the database using the `Product` model defined	N/A
The Product model	One or more of your fields did not match on your Product model. • `product_id` *(Must be uniquely auto-generated)* • `product_name` • `product_quantity` • `product_price`	The Product model should have 5 columns: • `product_id` • `product_name` • `product_quantity` • `product_price` **NOTE**: The value for each `product_id` should be a **unique** value that is auto-generated by Peewee.	N/A
Add existing data to database	The data from the `inventory.csv` file did not add all products and/or added products with incorrect values. • `product_quantity` as an Integer • `product_price` as an Integer • `date_updated` as a date	The app successfully adds each product contained in the `inventory.csv` file to the database, with the correct value in the correct column. • `product_quantity` as an Integer • `product_price` as an Integer • `date_updated` as a date	When entering products into the database, if a duplicate product name is found, the app will check to see which product entry was most recently updated and only save that data.

	Needs Work	**Meets Expectations**	**Exceeds Expectations**
Menu prompts	The menu did not contain one or more of the following menu options : • "View a single product's inventory" • "Add a new product to the database" • "Make a backup of the entire inventory" Allowing input from the user as: v, a, or b, respectively.	When the program is run: The user should be prompted with a simple menu of the following options: • "View a single product's inventory" • "Add a new product to the database" • "Make a backup of the entire inventory" **NOTE**: The user will be able to make this selection by pressing a single character (v, a, or b, respectively)	If the user enters any character other than v, a, or b then the app will notify them of their error and they can try again.
Menu option: v	Entering v at the menu prompt did not perform the correct actions to display the proper data and/or the database was not queried.	When the user enters v, the User is prompted to enter a product ID number, and after doing so, a human readable print of the correct product's information will be displayed in the terminal. **NOTE**: Product information data should always be queried from the database.	If the user enters a product ID that does not exist, a human readable error message will be displayed, and the user will be prompted to try again.
Menu option: a	Entering a at the menu prompt did not perform one or more of the correct actions: • input the Product Name • input the Product Quantity • input the Product Price • successfully save to the database	When the user enters a, they are prompted to enter a product's name, then the product's quantity, then the product's price. This product will then be successfully entered into the database.	When entering this product into the database, if a duplicate product name is found, the app will check to see which product entry was most recently updated and only save that data.

	Needs Work	**Meets Expectations**	**Exceeds Expectations**
Menu option: b	Entering `b` at the menu prompt did not perform one or more of the correct actions: • A CSV file named `backup.csv` was generated and stored in the project folder. • The CSV contents were not correct.	When the user enters `b`: Create a csv file called `backup.csv`. It should contain the current contents of the products database table. **NOTE**: Each product must be on its own line, with each field value separated by a comma. This file must be generated into the project folder.	The `backup.csv` file contains a header row with the Product field titles.