**CSV to Database**

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.

|  |  |
| --- | --- |
| **Database setup**   * A Sqlite database named inventory.db was created * The products table was created in the database using the Product model defined |  |

|  |  |
| --- | --- |
| **The Product model** |  |

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.

**Add existing data to database**

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.

|  |  |
| --- | --- |
| **Menu prompts** |  |

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** |  |

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** |  |

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.

|  |  |
| --- | --- |
| **Menu option: b** |  |

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.