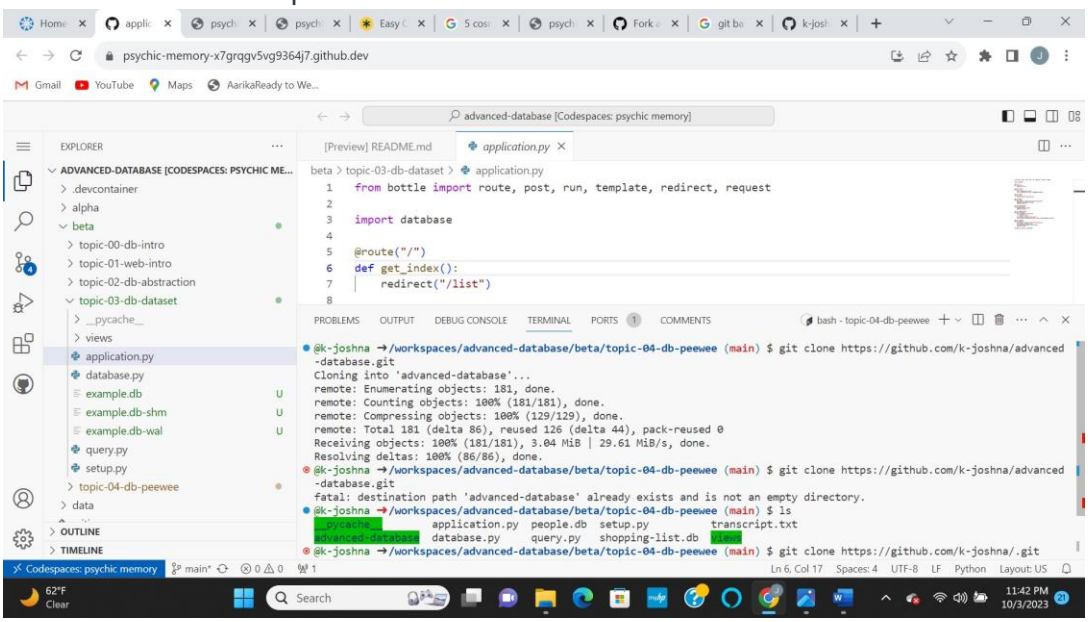
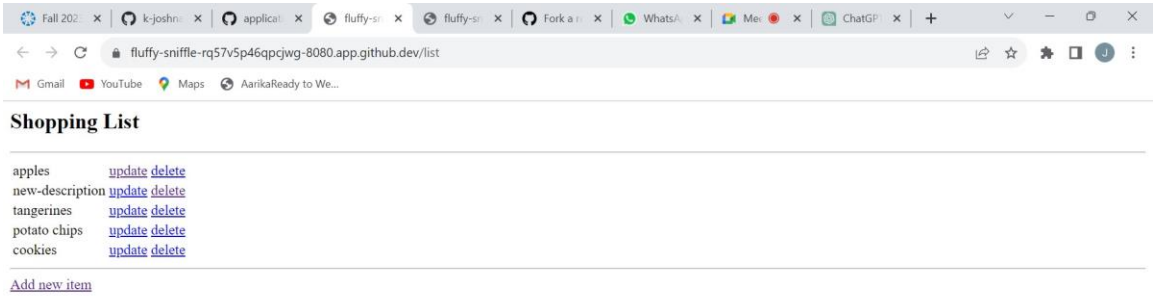
Homework 2

Joshna katta.

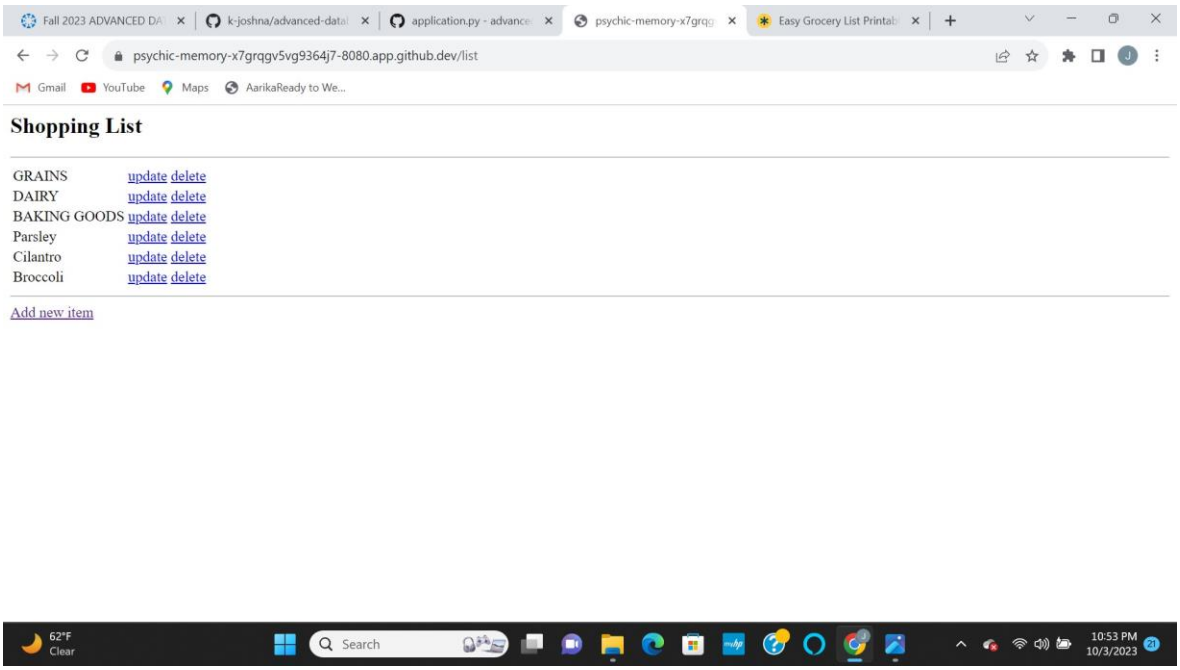
1. Fork the repo or clone the class repo, and get the code for the class web application to your own machine or codespace. Provide a screenshot of that.



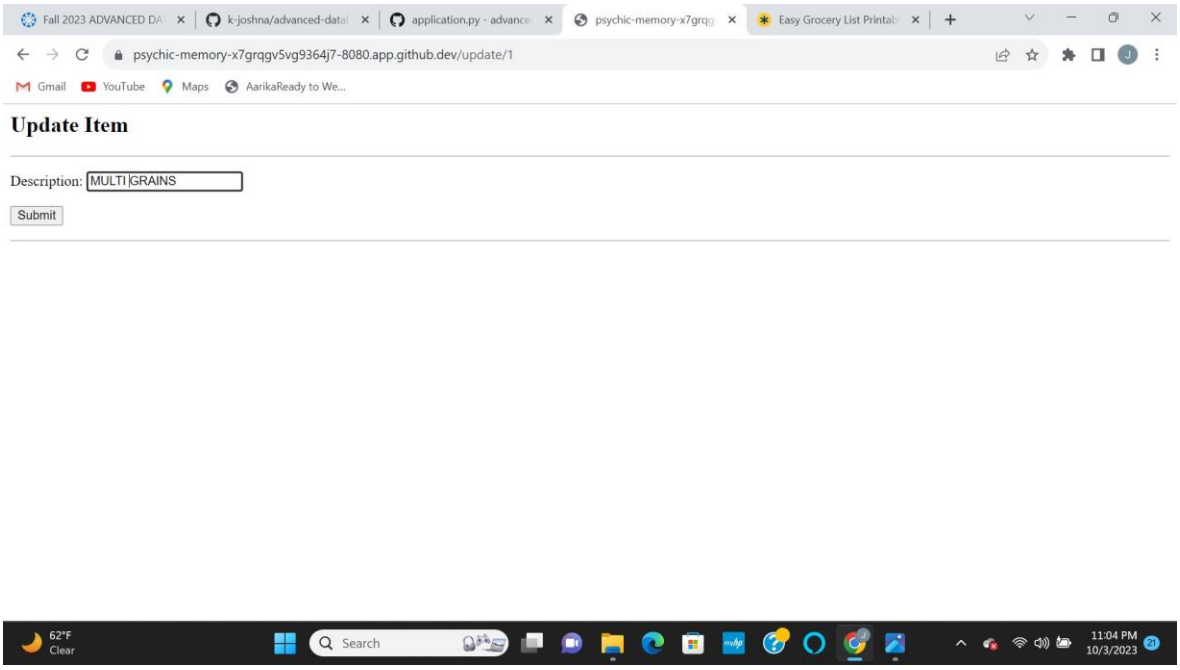
2. Get the version running that uses SQLite and the DB-API directly. Provide a screenshot of each the CRUD activities. 2.1 Read

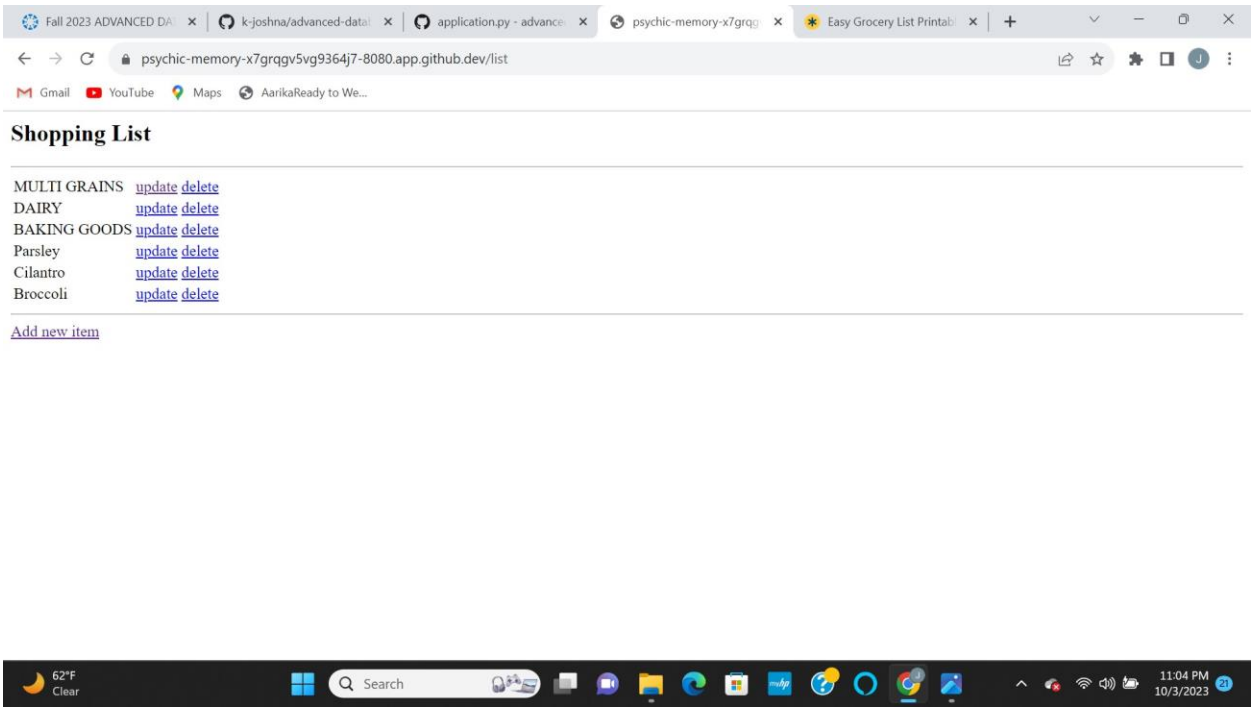


2.2Adding new items

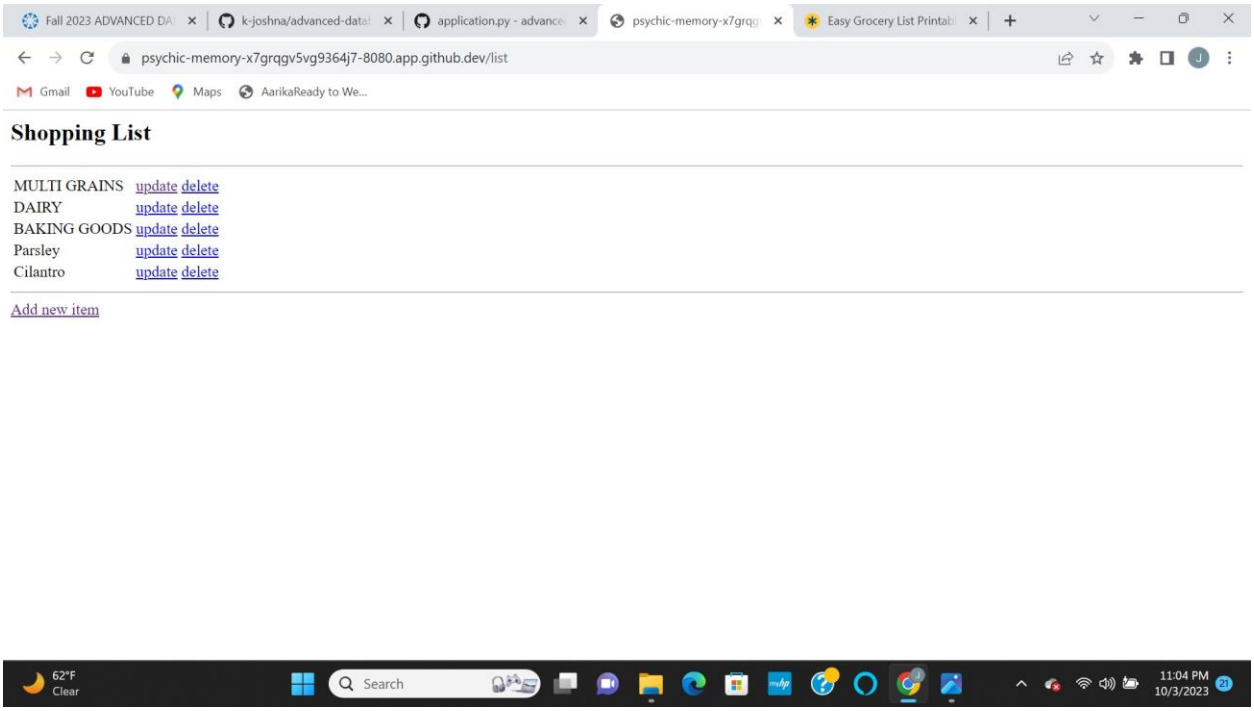


2.3 Updating Grains to multi-grains



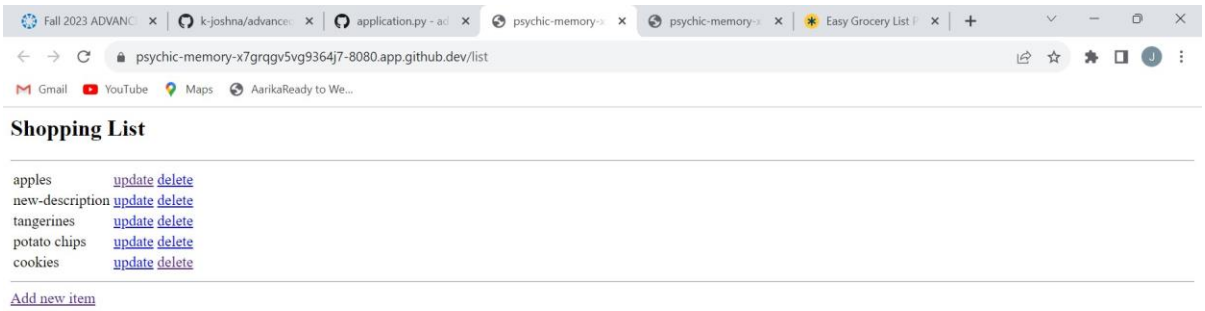


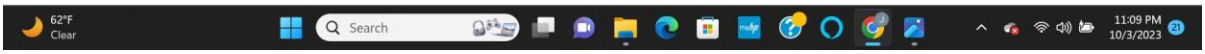
2.4 Delete Broccoli



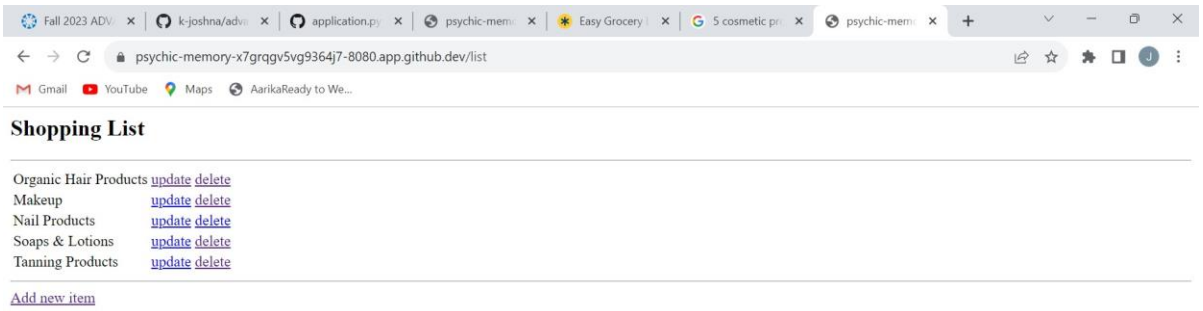
3. Get the version running that uses the PeeWee ORM. Provide a screenshot of each the CRUD activities

3.1 Read

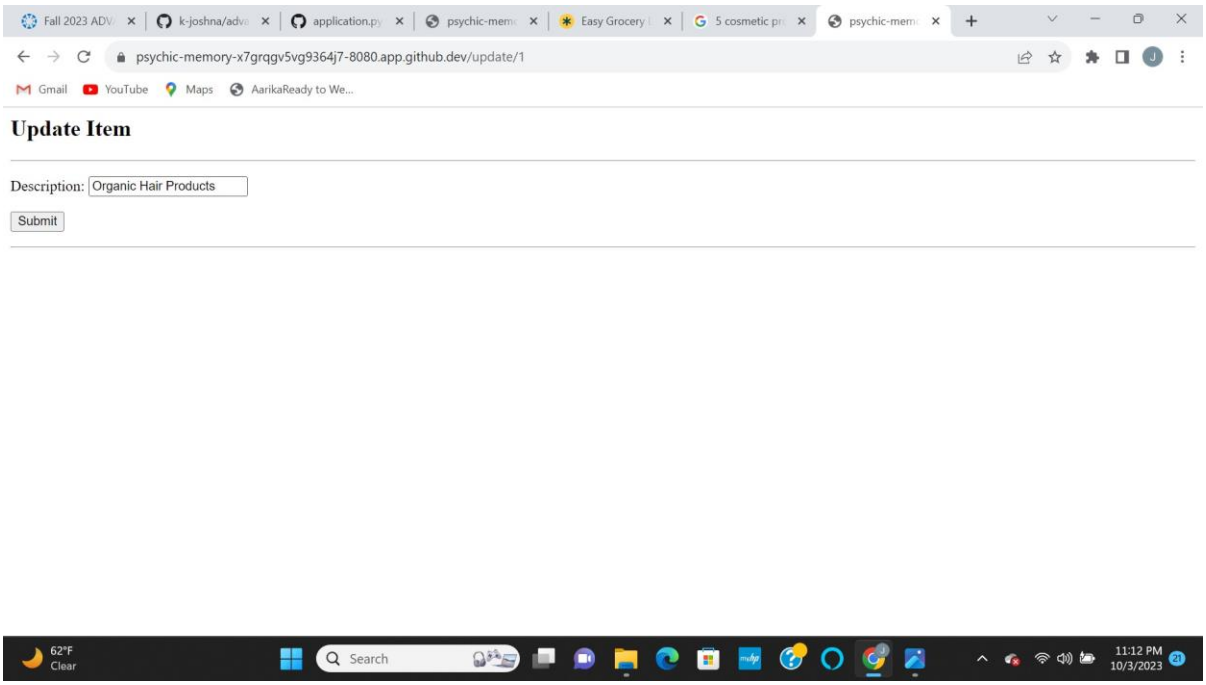




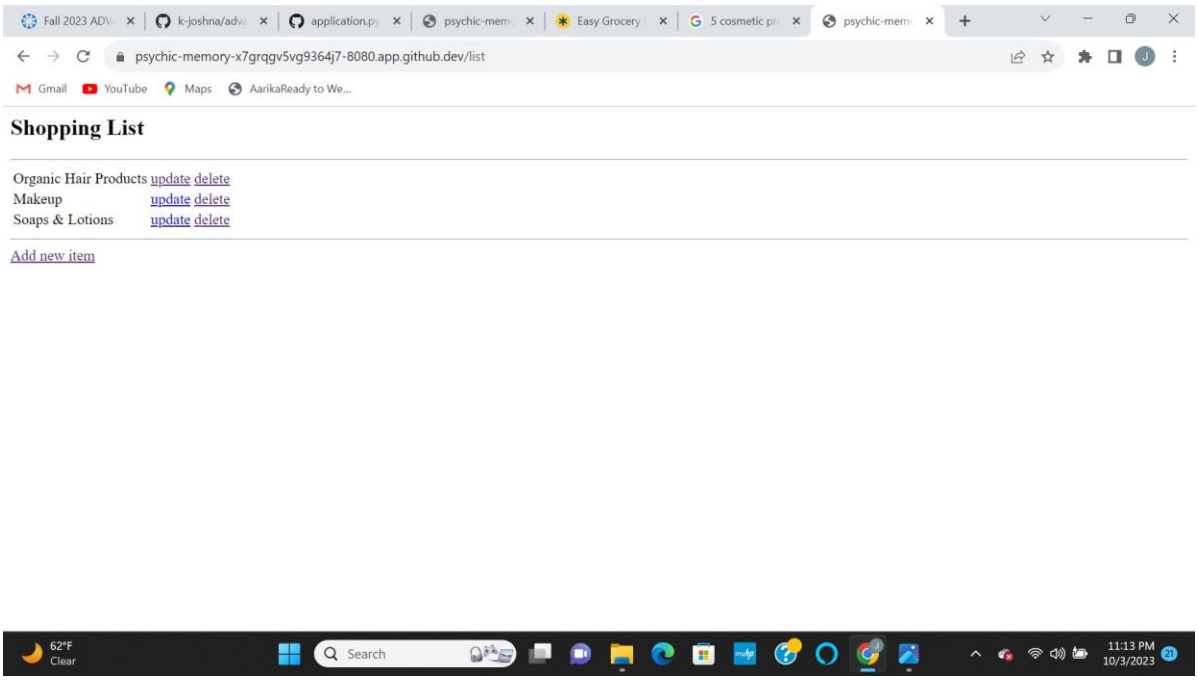
3.2 adding new items to the list



3.3 updating Organic hair products.



3.4 Deleting Nail and Tanning Products



4. Of the various technologies used to address the database, provide pros and cons for each one.

SQLite with DB-API:

Pros:

Lightweight: SQLite is a self-contained, serverless, and file-based database system. It's incredibly lightweight and suitable for small to medium-sized applications.

Zero Configuration: SQLite doesn't require complex setup or server configuration. It's easy to use and doesn't involve setting up a separate database server.

Cross-Platform: SQLite is cross-platform and works seamlessly on various operating systems, including Windows, macOS, and Linux.

Embedded: SQLite databases can be embedded directly into applications, making it an excellent choice for desktop and mobile applications.

SQL Compatibility: SQLite supports standard SQL syntax, making it easy to write SQL queries and work with existing SQL knowledge.

Cons:

Concurrency: SQLite is not well-suited for high-concurrency scenarios because it uses file-level locking, which can lead to contention in multi-user environments.

Limited Scalability: While great for small to medium-sized applications, SQLite may not scale well for large and complex systems.

Lack of Advanced Features: SQLite lacks some advanced database features like stored procedures and complex access control, which are available in larger database systems.

Peewee ORM:

Pros:

Abstraction: Peewee provides a high-level object-oriented abstraction over the database, making it easier to work with databases without writing raw SQL.

ORM Features: It offers features like model definition, query building, and relationships between tables, which simplify database operations and maintainability.

Database Agnostic: While Peewee provides support for SQLite, it's also compatible with other relational database systems like MySQL, PostgreSQL, and Oracle.

Concurrency: Peewee offers built-in support for handling database concurrency and transactions, making it suitable for multi-user applications.

Ease of Use: Peewee's simple and intuitive syntax reduces the learning curve for developers, especially those new to database programming.

Cons:

Learning Curve: While Peewee simplifies database operations, there is still a learning curve associated with using an ORM, especially for complex database structures and queries.

Performance Overhead: In some cases, using an ORM like Peewee can introduce a slight performance overhead compared to writing optimized raw SQL queries.

Complexity: For simple projects, introducing an ORM like Peewee may add unnecessary complexity.