**Description:** The purpose of the Mini Project is to reinforce skills that have been covered in recent modules. Select a data set and analyse it using the techniques learned in recent modules. Prepare and present a 10-minute presentation in class.  
  
----------- **Rubric:**

* Explanation of the business context behind the dataset
* Quality of presentation - engaging, well structured, not too short or too long
* Quality of the notebook - code well documented, runs correctly

-------------------------------------------------------------------------------------------------------------  
  
Mini Project 3 -   
  
expectation and AC  
  
1.) Have a topic where you are solving a data service problem, have a reason why you’d want to store things in a database, why you wouldn’t just let that be managed by the browser/session.  
2.) design a dB (SQL or NO-SQL)   
3.) build a dB structure (collections/tables/models/schema)  
4.) show how you’d query your dB to get your data  
5.) connect it to a simple API which is just a series of routes that will deliver meaning content to whoever uses it.  
  
tldr:  
pick a dbms  
create express api app that connects to that dbms  
show solutions by running API calls.  
(Swagger is a plus)

Here’s the plan for Mini Project 3:

**Objective:**

Build an Express API that connects to a database (SQL or NoSQL) to solve a data service problem.

**Steps to Follow:**

1. **Define the Problem**
   * Pick a topic where managing data in a database is necessary rather than relying on browser storage or sessions.
   * Explain why persistence is required beyond a single user session.
2. **Choose a Database**
   * Select either **SQL (e.g., MySQL, PostgreSQL)** or **NoSQL (e.g., MongoDB, Firebase, DynamoDB)** based on the needs of your project.
3. **Design the Database**
   * Define collections (NoSQL) or tables (SQL).
   * Create a schema/model outlining how data will be structured.
4. **Build the Database Structure**
   * Implement tables (for SQL) or collections (for NoSQL).
   * Set up relationships or references if needed.
5. **Query the Database**
   * Write queries to retrieve, update, insert, and delete records.
   * Use SQL queries (e.g., SELECT, INSERT, UPDATE, DELETE) or NoSQL methods (e.g., MongoDB’s find, aggregate, etc.).
6. **Connect to an Express API**
   * Develop an API with endpoints to interact with the database.
   * Use RESTful principles to expose meaningful data.
   * Include CRUD operations: Create, Read, Update, Delete.
7. **Test API Calls**
   * Use tools like **Postman** or **Thunder Client** to validate endpoints.
   * Swagger documentation is a bonus for API reference.

**End Goal:**

A functioning API with a structured database, connected via Express, capable of handling requests efficiently.