A blue and gold text on a black background

Description automatically generated

**Medicare Pharmacy Sales Analysis**

Project Report Submitted in partial fulfilment of the requirement for the award of degree of

**Master of Science (Data Science)**

**A logo with a star and a candle

Description automatically generated**

Submitted By

**NAME: Saail D. Tayshete**

**REG NO: 2348452**

Submitted To

**Dr. Gobinath R**

**DEPARTMENT OF COMPUTER SCIENCE**

**CHRIST (DEEMED TO BE UNIVERSITY), Bangalore Yeshwanthpur Campus, Karnataka-560073**

**# Background:**

****

Medicare Pharmacy aims to enhance its sales strategy by analyzing customer transactions and identifying areas with varying sales performances. The project focuses on leveraging data collected from the website to gain insights into customer preferences and sales patterns based on geographical locations (identified by pincode).

**1. Aim of the Project:**

**The primary goal of the project is to assess and analyze the sales performance of Medicare Pharmacy in different geographical areas. Specifically, the project aims to:**

* **Collect customer transaction data through a website form.**
* **Store the collected data in MongoDB for efficient data management.**
* **Analyze sales data to identify areas with the highest and lowest sales based on pincode.**
* **Provide actionable insights to the company for strategic decision-making.**

**2. Project Components:**

**2.1. Website Development:**

* Developed a dynamic website using Node.js, JavaScript, and HTML to serve as the user interface for customer interactions.
* Implemented a form on the website to capture essential customer information, including name, product name, email, pincode, mobile number, and product quantity.

2.2. Database Integration:

* Established a connection between the website and MongoDB to store customer data.
* Utilized JavaScript to handle data input from the website form and store it in the MongoDB database.
* Ensured proper validation and handling of user inputs to maintain data integrity.

2.3. Sales Data Analysis:

* Implemented a data analysis module to assess the sales performance based on pincode.
* Calculated aggregate sales data, including total sales, average sales, and product quantities for each pincode.

2.4. Identification of High and Low Sales Areas:

* Utilized the sales data analysis to identify areas (pincodes) with the highest and lowest sales.
* Generated reports or visualizations to present the findings in a clear and understandable format.

**3. Steps and Execution:**

**3.1. Website and Database Setup:**

* Developed the website using HTML for the user interface.
* Used Node.js for server-side scripting to handle requests and responses.
* Integrated MongoDB to store customer data.

**3.2. User Interaction:**

* Designed and implemented a user-friendly form on the website to collect relevant customer details.

**3.3. Database Interaction:**

* Implemented JavaScript functions to handle user inputs and store them in the MongoDB database.
* Ensured data validation to prevent erroneous or incomplete entries.

**3.4. Sales Data Collection:**

* Set up scripts to collect and aggregate sales data based on pincode.
* Implemented error handling to address potential issues in data collection.

**3.5. Analysis and Reporting:**

* Analyzed the collected data to identify areas with the highest and lowest sales.
* Generated reports or visualizations to present the findings.

**3.6. Results and Insights:**

* Presented clear insights into the areas with the highest and lowest sales based on pincode.
* Provided recommendations for strategic interventions in areas with lower sales to improve overall sales performance.

**4. Data:**

Here we are getting orders of medicines(inputs) form users for a pharmacy website.

1. User Input Form Description:
2. Customer Name (name):

Type: Text

Description: Full name of the customer.

1. Product Name (productName):

Type: Text

Description: Name of the product being purchased.

1. Quantity (quantity):

Type: Number

Description: Quantity or number of units being purchased.

1. Email (email):

Type: Email

Description: Valid email address for communication and confirmation.

1. Mobile Number (mobileNumber):

Type: Tel

Description: 10-digit mobile or phone number.

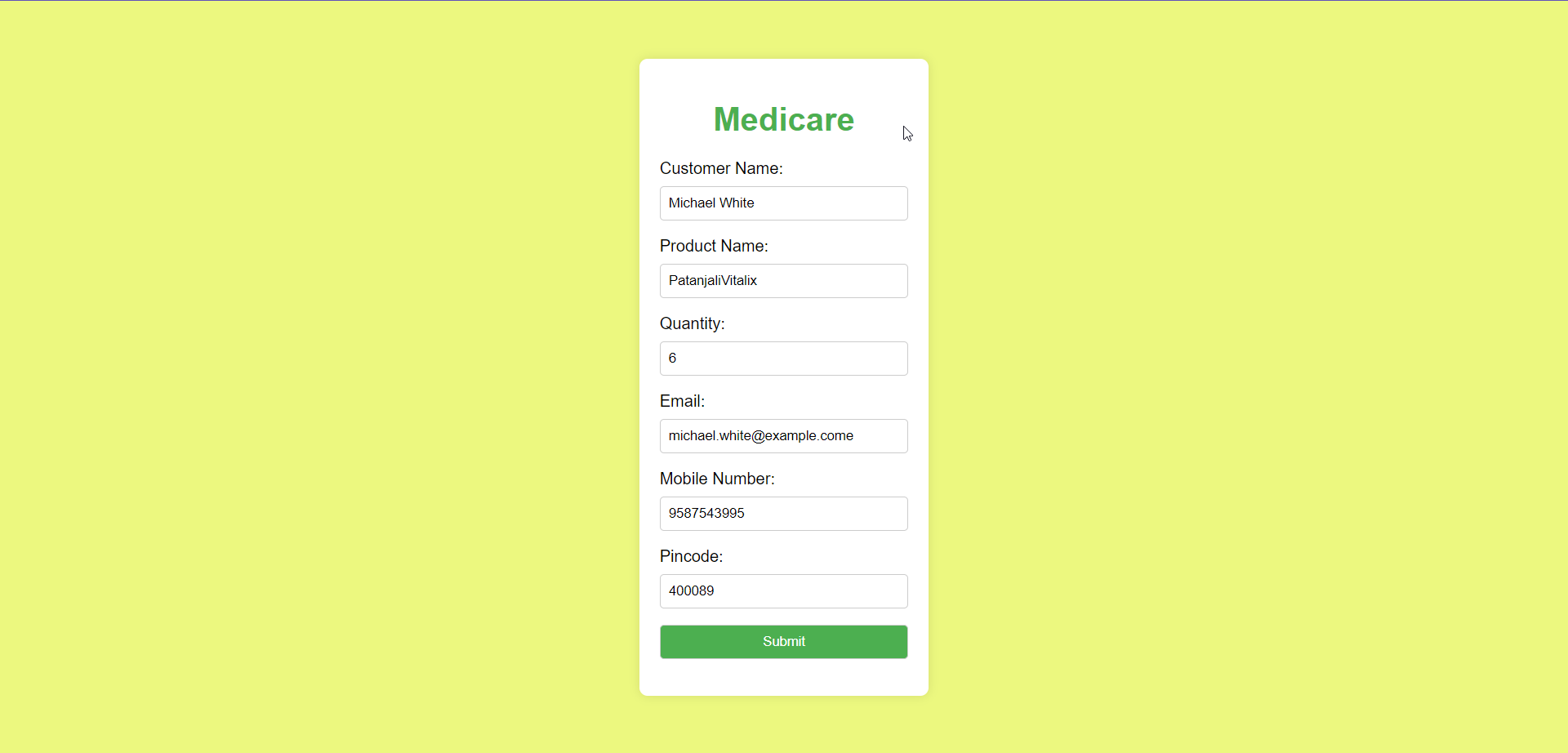
1. Pincode (pincode):

Type: Text

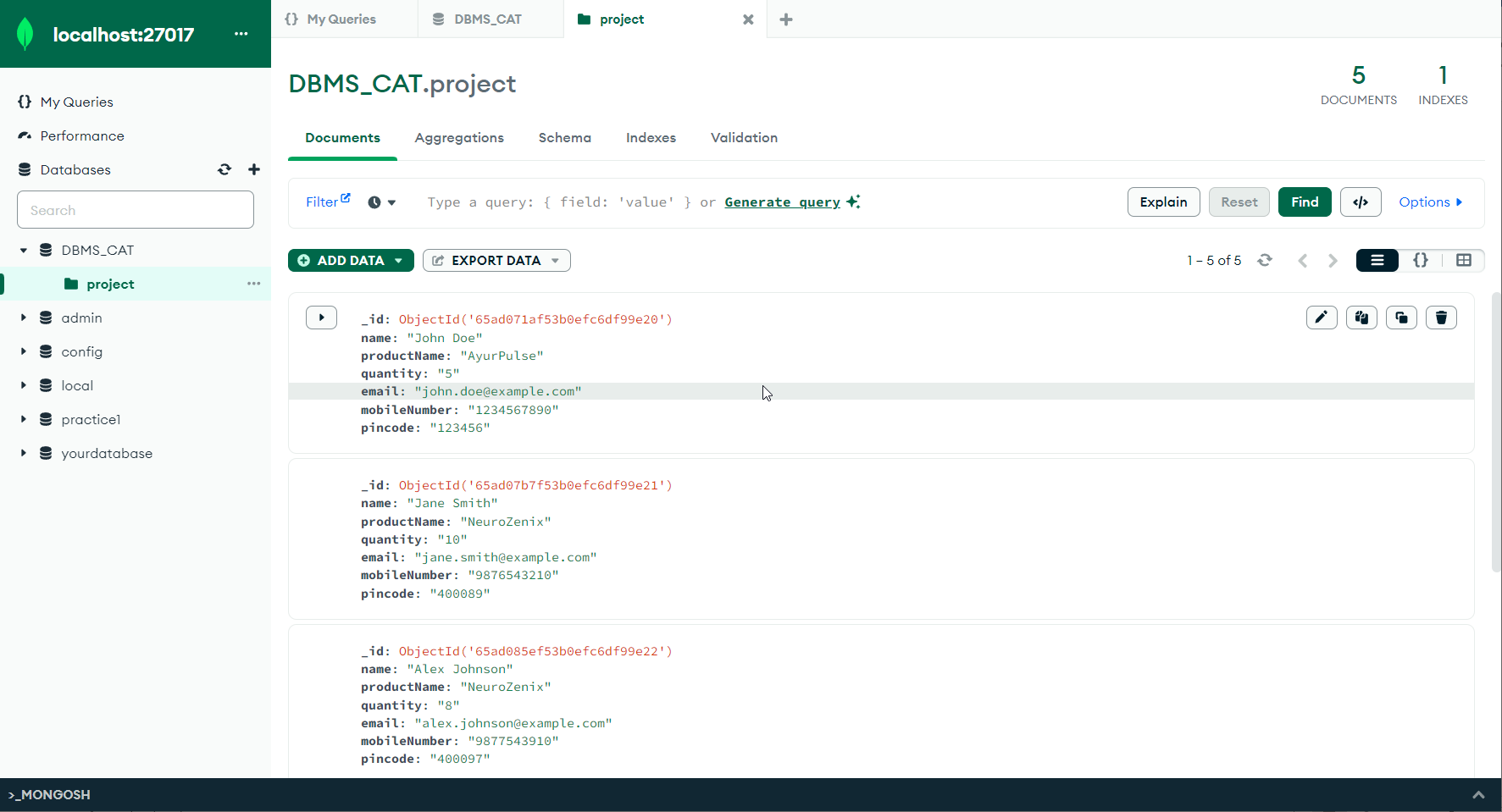
Description: 6-digit postal code or pin code.

**5.Connecting MongoDB to a Website using Node.js**

* **Install Node.js and npm:**
  + Ensure Node.js and npm (Node Package Manager) are installed on your system. You can download them from <https://nodejs.org/>.
* **Initialize Your Node.js Project:**
  + Create a new directory for your project and navigate to it using the terminal or command prompt. Run npm init to initialize a new Node.js project. Follow the prompts to set up your package.json file.
* **Install Required npm Packages:**
  + Install the necessary npm packages for your project:
  + Cmd: **npm install express mongoose**
* **Set Up Express Application:**
  + Create an app.js file or your main server file and set up an Express application. Import Express and create an instance of the app.
* **Connect to MongoDB:**
  + Use Mongoose, an ODM (Object Data Modeling) library for MongoDB and Node.js, to connect to your MongoDB database. Require and configure Mongoose in your app.js file.
  + **JS: const mongoose = require('mongoose');**
  + **mongoose.connect('mongodb://localhost:27017/your-database-name', {**
  + **useNewUrlParser: true,**
  + **useUnifiedTopology: true,**
  + **});**
  + **const db = mongoose.connection;**
  + **db.on('error', console.error.bind(console, 'MongoDB connection error:'));**
  + **db.once('open', () => {**
  + **console.log('Connected to MongoDB');**
  + **});**
* **Define MongoDB Schema and Model:**
  + Define a schema for your MongoDB data and create a model using Mongoose. This step helps structure the data you'll be storing.
  + **JS: const { Schema, model } = mongoose;**
  + **const userSchema = new Schema({**
  + **name: String,**
  + **email: String,**
  + **// ... define other fields as needed**
  + **});**
  + **const User = model('User', userSchema);**
* **Create Routes and Handle MongoDB Operations:**
  + Set up routes in your Express application to handle CRUD operations with MongoDB. Use the defined model to perform database operations.
  + **JS: app.post('/submitData', (req, res) => {**
  + **const { name, email, /\* other fields \*/ } = req.body;**
  + **const newUser = new User({ name, email /\* other fields \*/ });**
  + **newUser.save((err) => {**
  + **if (err) {**
  + **res.status(500).send('Error saving to database');**
  + **} else {**
  + **res.status(200).send('Data saved to database');**
  + **}**
  + **});**
* **Start Express Server:**
  + Start your Express server and listen on a specific port.
  + **JS: const port = process.env.PORT || 3000;**
  + **app.listen(port, () => {**
  + **console.log(`Server is running on port ${port}`);**
  + **});**
* **Run Your Node.js Application:**
  + In the terminal, run your Node.js application using:
  + **Cmd: node app.js**
* **Created Medicare website:**

****

**MongoDB Data:**

****

**CRUD:**

* Update Data:

Select the document you want to update.

Click on the "Edit Document" button.

Modify the fields as needed and click "Save Changes."

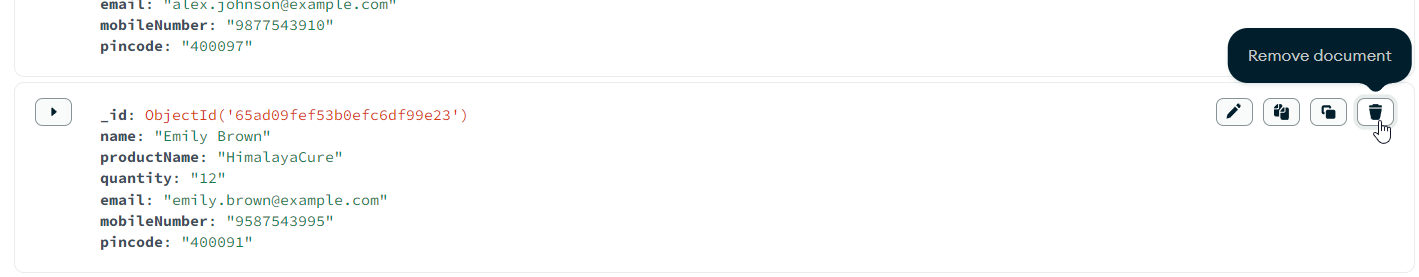


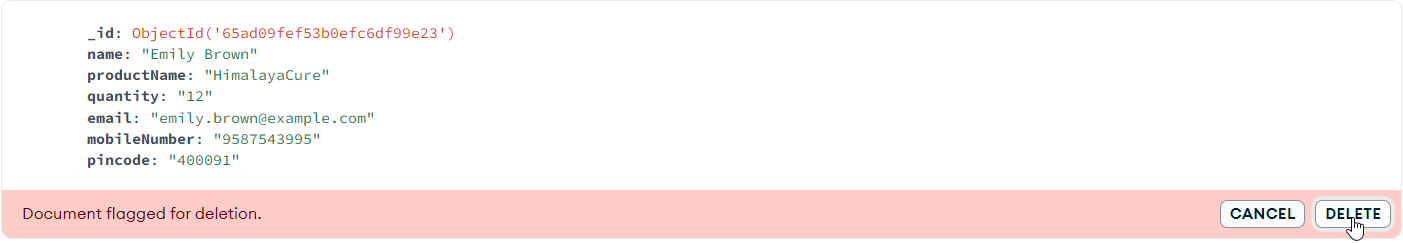
* Delete Data:

Select the document you want to delete.

Click on the "Delete Document" button.

Confirm the deletion when prompted.

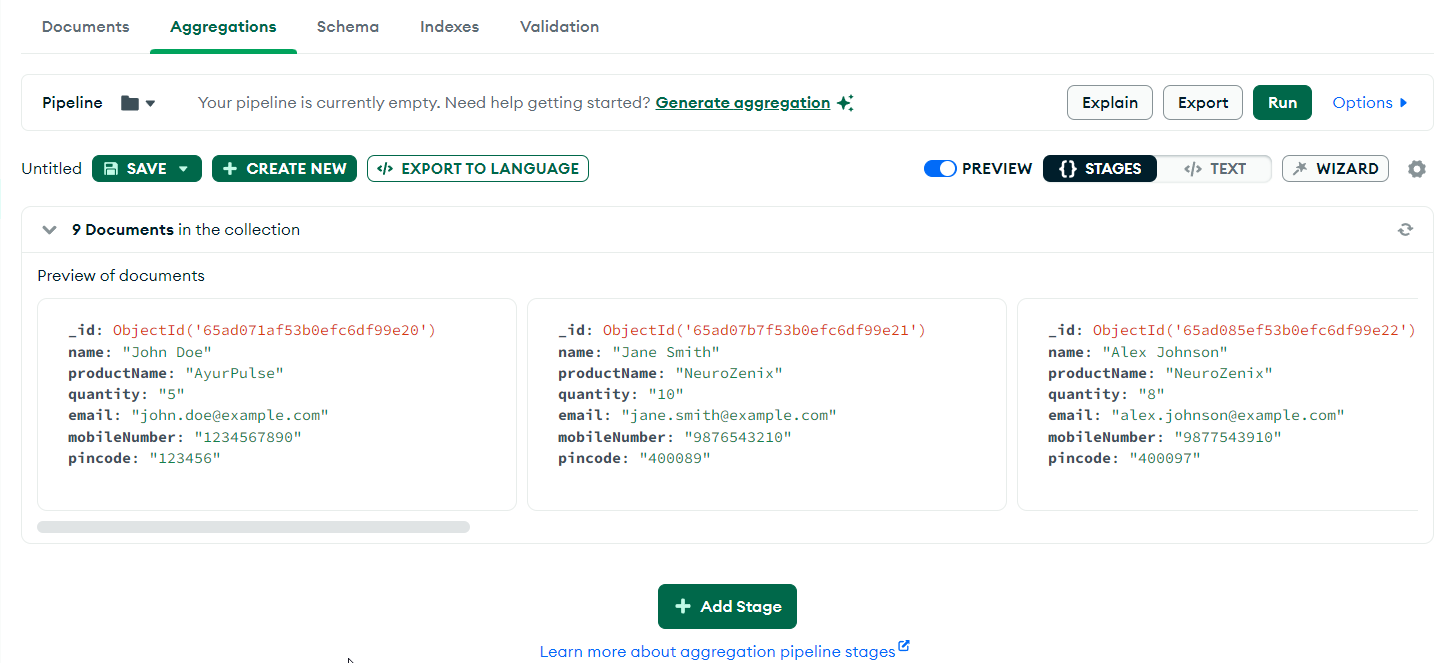




* Aggregation (Optional):

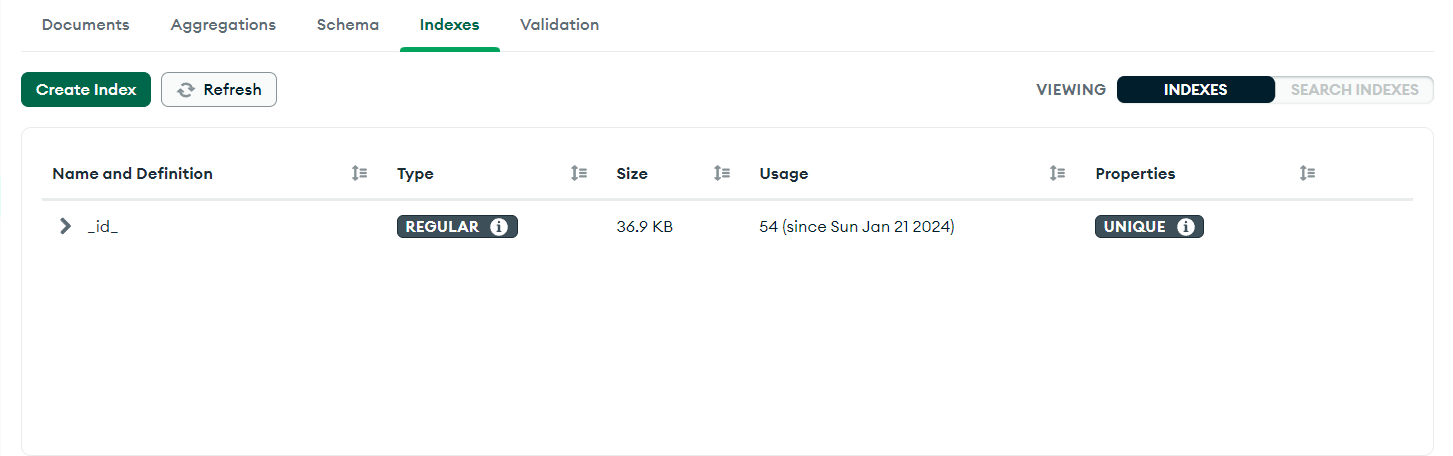
MongoDB Compass provides an Aggregation Pipeline Builder for complex queries.

Navigate to the "Aggregations" tab to build and execute aggregation pipelines.



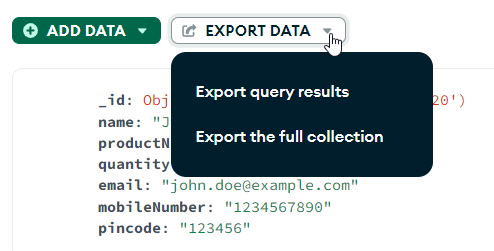
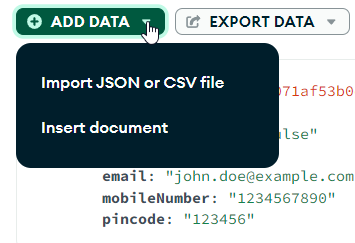
* View Indexes:

In the "Collection" tab, click on the "Indexes" tab to view and manage indexes for better performance.



* Export and Import Data:

Use the "Export" and "Import" options to move data in and out of collections.



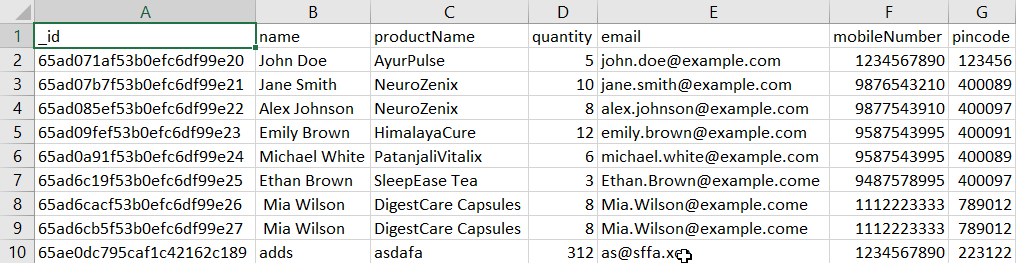
* Query Performance (Optional):

Analyze query performance using the "Performance Advisor" tab to optimize your queries.

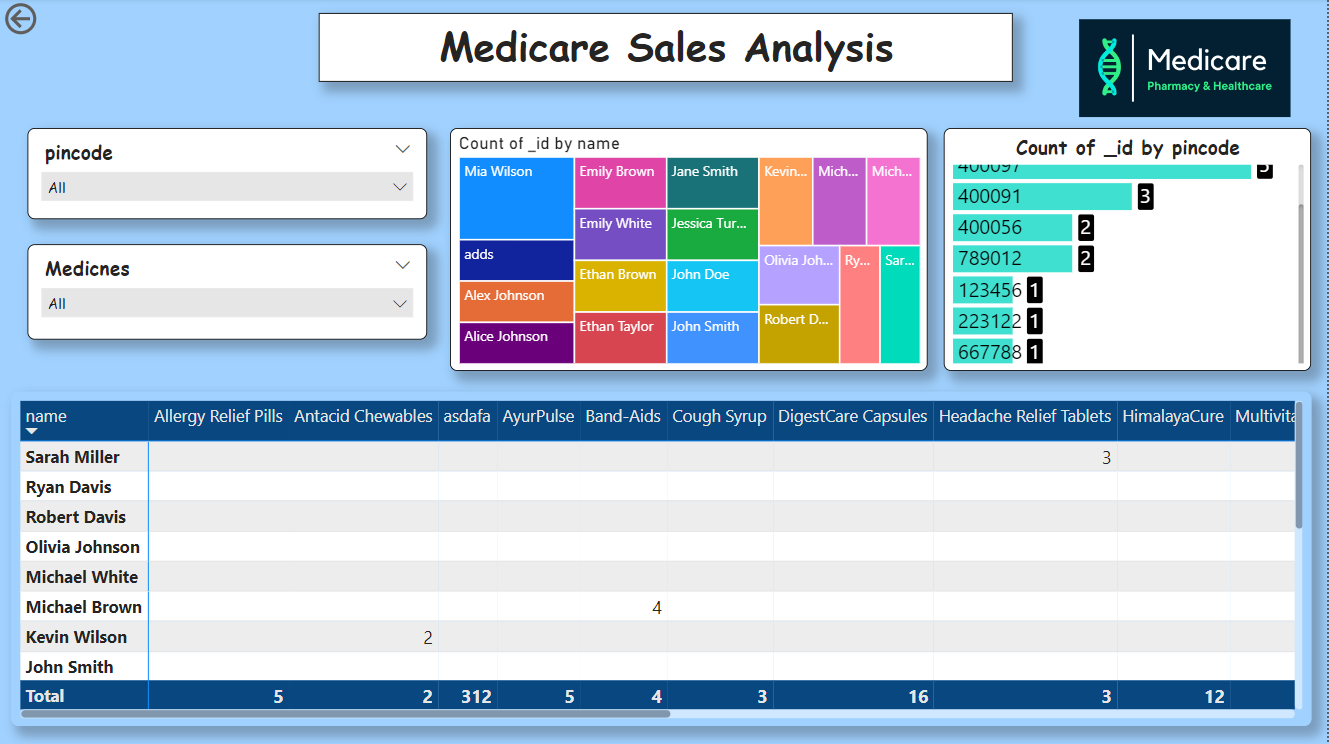
* Manage Database Users (Optional):

In the "Database" tab, manage database users and roles under the "Security" section.

**CSV file:**

****

**Power BI Sales Analysis Report:**

****

**1. Introduction:**

**1.1 Purpose:**

The Power BI Sales Analysis focuses on evaluating the sales data acquired from the Medicare Pharmacy website. The data, sourced from user inputs on the website and stored in MongoDB, has been imported into Power BI for in-depth analysis.

**2. Key Observations:**

**2.1 Areas with Lowest Sales:**

Observation: The areas with the lowest sales are identified by the pincode 667788 and 223122. Insight: Further investigation into these areas can provide insights into potential factors contributing to lower sales, enabling targeted strategies for improvement.

**2.2 Customer Purchase Insights:**

Observation: The Power BI dashboard allows for a detailed breakdown of customer names and the quantity of different medicines they have ordered. Insight: Understanding customer preferences and purchase patterns can guide inventory management and marketing efforts.

**2.3 Medicines Supplied in Different Areas:**

Observation: Through chart filtration, the Power BI analysis displays the names of different medicines supplied in various areas. Insight: Identifying the demand for specific medicines in different regions enables optimized stock management and tailored marketing strategies.

**2.4 Regular Customer Identification:**

Observation: The Power BI dashboard provides information on regular customers visiting the website. Insight: Recognizing and rewarding regular customers can foster customer loyalty and contribute to repeat business.

**Findings:**

**1. Identification of Areas with Lowest Sales:**

Power BI's pincode analysis revealed specific regions with the pincodes 667788 and 223122 that exhibited the lowest sales. This identification serves as a critical insight for strategic decision-making.

**2. Customer Purchase Patterns:**

Power BI's detailed breakdown of customer names and the quantity of different medicines ordered provided nuanced insights into customer preferences. Understanding these patterns is vital for inventory management and personalized marketing.

**3. Medicine Distribution by Region:**

Utilizing filtered charts in Power BI, the project successfully showcased the names of medicines supplied in different areas. This granularity aids in optimizing regional inventory and tailoring product offerings to meet specific demands.

**4. Regular Customer Identification:**

Power BI's capabilities allowed for the identification of regular customers, presenting an opportunity to implement targeted loyalty programs and promotions. Recognizing and rewarding loyal customers can significantly impact customer retention.

**Detailed Conclusion:**

The Medicare Pharmacy website project not only achieved its primary aim of identifying regions with the lowest sales but also provided a comprehensive understanding of customer behavior and regional demand. The project successfully integrated web development, database management, and data visualization tools for a holistic approach.

**Future Recommendations:**

1. **Continuous Monitoring:**
   * Implement a system for ongoing monitoring of sales data to identify emerging trends and areas for improvement. Regular analysis ensures the adaptation of strategies to evolving customer preferences.
2. **Enhanced Customer Engagement:**
   * Introduce features on the website to encourage customer engagement, such as personalized promotions, newsletters, or loyalty programs. Engaged customers are more likely to become regular patrons.
3. **Collaboration with Marketing Strategies:**
   * Collaborate the findings with marketing strategies to implement targeted campaigns in regions with lower sales. Alignment between data insights and marketing efforts enhances the overall effectiveness.
4. **Feedback Mechanism:**
   * Implement a customer feedback mechanism on the website to gather insights directly from users. Understanding customer feedback is crucial for continuous improvement and maintaining customer satisfaction.

**References:**

1. **Documentation and Official Websites:**
   * **MongoDB Documentation: Official documentation for MongoDB, offering comprehensive guides, tutorials, and references.**
   * **Node.js Documentation: Official documentation for Node.js, providing in-depth information on its features and capabilities.**
2. **Web Development Frameworks:**
   * [**Express.js Documentation**](https://expressjs.com/)**: Documentation for Express.js, a popular web application framework for Node.js.**
   * **Bootstrap Documentation: Bootstrap documentation for creating responsive and mobile-first websites.**
3. **Learning Platforms:**
   * **MDN Web Docs: Mozilla Developer Network offers detailed web development documentation and guides.**
   * [**W3Schools**](https://www.w3schools.com/)**: W3Schools provides tutorials and references on various web development technologies.**
4. **Node.js and MongoDB Tutorials:**
   * **Node.js + MongoDB Tutorial: Tutorial on using Node.js with the Express framework.**
   * **MongoDB and Node.js Integration: MongoDB's official guide on using Node.js with MongoDB.**
5. **Web Security Resources:**
   * [**OWASP**](https://owasp.org/)**: The Open Web Application Security Project provides resources on web application security best practices.**
   * **Express.js Security Best Practices: Security best practices specific to Express.js.**
6. **Code Version Control:**
   * **Git Documentation: Documentation for Git, a version control system widely used in software development.**
   * **GitHub Guides: Guides on using GitHub for collaborative development.**
7. **Healthcare Data and Regulations:**
   * **HIPAA Security Rule: Health Insurance Portability and Accountability Act (HIPAA) security rule for handling healthcare data.**
8. **Online Communities and Forums:**
   * [**Stack Overflow**](https://stackoverflow.com/)**: Community-driven platform for asking and answering programming-related questions.**
   * **Node.js Google Group: Node.js discussion group for seeking help and insights.**

**Top of Form**