

Assignment 4

Developing an Analytical Platform Using Node.js and MongoDB

Overview:

You are tasked with creating an analytical platform that retrieves data from a MongoDB database and visualizes it in the form of time-series graphs and other statistical measures. The platform will allow users to select which data field they wish to analyze. It should offer multiple visualization options, like line graphs, bar charts, and statistical summaries for the selected data fields over time.

Task	Description	Points
Backend Development	Develop a backend service using Node.js with the Express.js framework to interact with a MongoDB database.	15%
Data Storage & Modeling	Store time-series data in MongoDB using a well-defined schema.	10%
API Endpoints	Implement RESTful API endpoints to fetch data, provide filtering options, and return relevant metrics.	25%
Data Visualization	Integrate a frontend and use an API client (e.g., Postman) to visualize data, utilizing libraries such as Chart.js or D3.js.	25%
Error Handling and Validation	Ensure proper error handling, input validation, and pagination when necessary.	15%
User Input	Allow users to select which field they want to visualize and specify time ranges for analysis.	10%

Requirements:

MongoDB Schema Design:

Create a collection named `measurements` where each document represents a single measurement with the following fields:

- `timestamp`: (Date) The time the measurement was recorded.
- `field1`: (Number) A numerical field (e.g., temperature, sales, stock prices).

- field2: (Number) Another numerical field (e.g., humidity, customer satisfaction, product rating).
- field3: (Number) Another numerical field (e.g., CO2 levels, units sold, website traffic).
- Additional fields can be added to suit your use case

Backend Development (Node.js/Express):

Set up a Node.js application with Express.js.

Implement a MongoDB connection using Mongoose.

Create RESTful API endpoints to:

- Fetch time-series data from the measurements collection.
- Allow filtering of data by date range (start_date and end_date).
- Select which field to visualize (e.g., field1, field2, or field3).
- Fetch additional metrics, such as averages, minimum, maximum, and standard deviation for the selected field.

Example Endpoints:

- GET /api/measurements?field=field1&start_date=2024-10-01&end_date=2024-10-20
- GET /api/measurements/metrics?field=field2

Frontend:

Create a simple frontend (using HTML/JavaScript) and utilize a tool like Postman to visualize the data in graphs.

Use Chart.js or D3.js to plot time-series graphs (e.g., line charts, bar charts) for the selected field.

Allow the user to input the time range and field selection through a form and fetch data via the API.

Visualization & Metrics:

Users should be able to visualize:

- Time-series data for any selected field.
- Calculate and display basic statistics such as:
 - Average of the field over the selected period.
 - Minimum and Maximum values.
 - Standard Deviation.

Display multiple visualization types (line graph, bar chart, etc.).

Error Handling & Validation:

Ensure that the API handles errors gracefully, including:

- Invalid field names.
- Invalid date formats.
- Nonexistent data in the specified range.

Implement input validation to avoid server crashes.

Note: Here are several free database sources where you can download real-world data for your analytical platform assignment

Dataset description	Data format	Examples
Kaggle provides a wide variety of free datasets for data science and machine learning projects. Categories include finance, healthcare, social sciences, and much more.	CSV, JSON, Excel	E-commerce transactions; Financial stock data; Weather data
The UCI Machine Learning Repository is one of the most popular sources of datasets for research and education. It offers well-organized datasets across multiple disciplines.	CSV, SQL, ARFF	Online Retail Dataset; Energy consumption dataset; Air quality data
Amazon Web Services (AWS) offers a variety of public datasets available for free usage. These include healthcare, climate, genomics, and geospatial data.		NOAA Weather Data; Landsat satellite imagery; U.S. Census data
Data.gov is a U.S. government website that provides free access to a massive range of public datasets across different sectors, including education, finance, health, and environment.	CSV, JSON, XML	U.S. crime data; Healthcare statistics; Transportation data
The World Bank provides free access to global development data, including economic,	CSV, XML, Excel	Global GDP data;

social, and environmental data from around the world.		Population growth data; Poverty rates
Google Dataset Search helps you find datasets stored across the web. It indexes both large public datasets and small ones. You can search for datasets by keywords.		Health and medical data; Financial and stock market data; Geospatial data
OpenStreetMap is a free, editable map of the world that provides geographical data that can be downloaded and used for a variety of purposes.	OSM, CSV, GeoJSON	Geolocation data; Road networks; Urban planning data
OpenWeatherMap provides free access to weather data from cities around the world. Developers can use their free API to gather real-time and historical weather data.	JSON, XML	Current and historical weather data; Forecast data; Air pollution data

How to Use These Sources:

- You can download datasets in formats such as CSV or JSON and write a Node.js script to import them into your MongoDB database.
- Many of these sources also offer APIs, allowing you to automate data collection and insertion into MongoDB using scheduled tasks (e.g., cron jobs) within their Node.js backend.