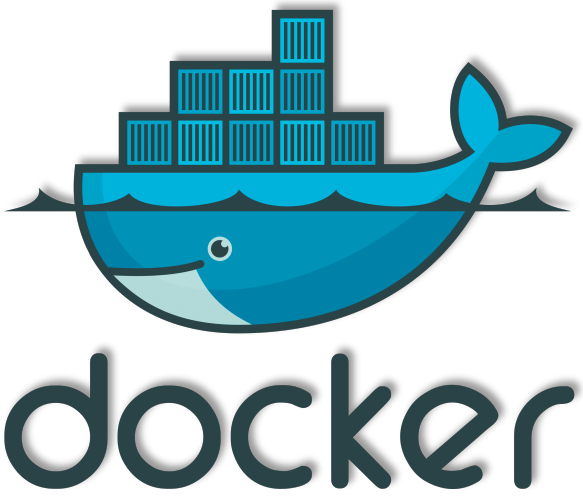
Containerization Assignment Report

Fares Qawasmi

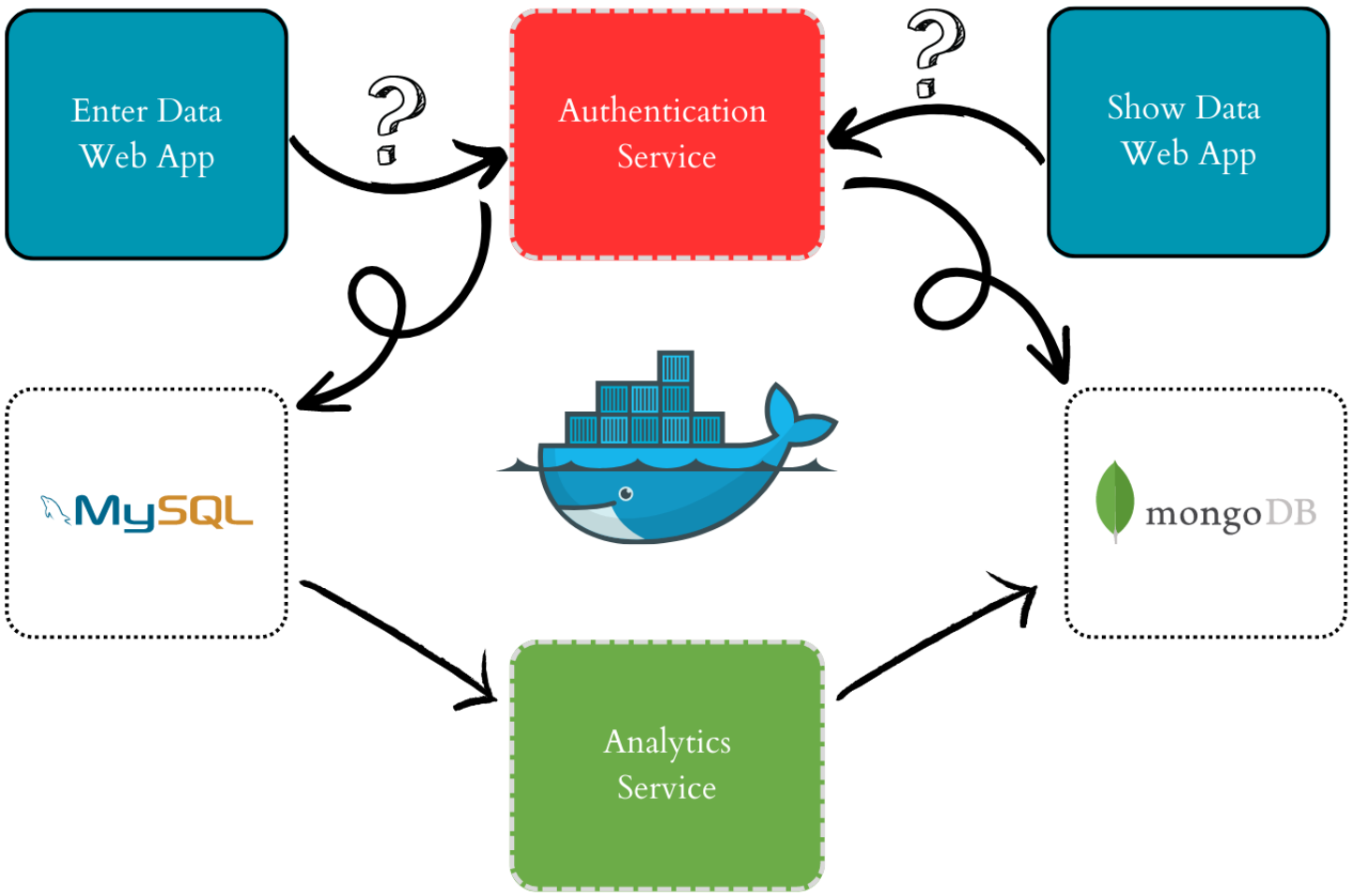
Dr. Motasem Aldiab



Abstract:

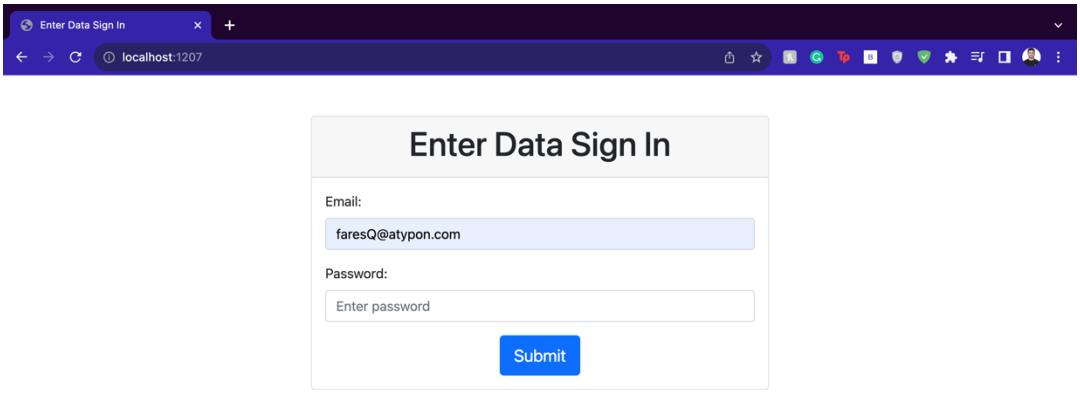
This report presents the design and implementation of a microservices-driven data collection and analytics system composed of distinct microservices, including the "Enter Data" and "Show Results" web applications, an Authentication Service, and an Analytics Service. The "Enter Data" web app permits users to input the temperature, authenticated through the Authentication Service, and stored in a MySql database. The Analytics Service extracts data from MySql, generates the maximum temperature, and stores results in Mongo DB. The "Show Results" app, also authenticated by the Authentication Service, retrieves the records from a Mongo DB service. Authentication Service ensures secure access to both apps.

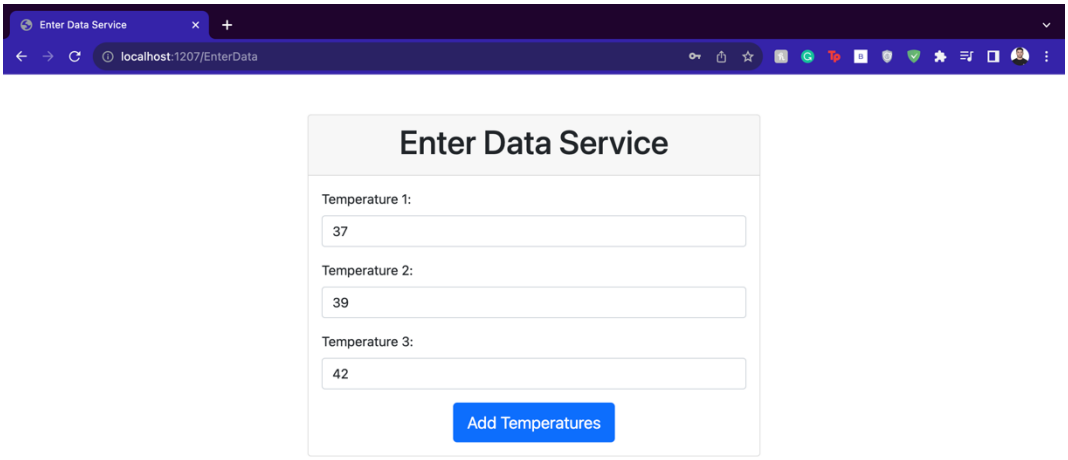
Demonstration:



Enter Data Web App:

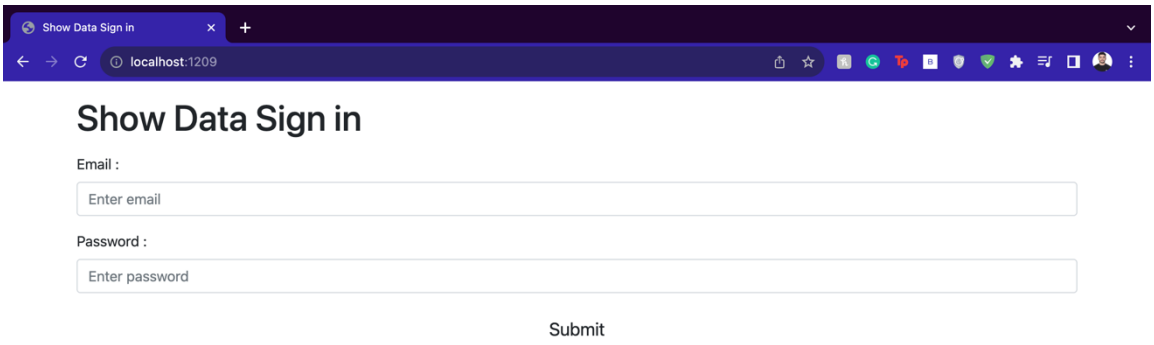
The Enter Data web app service is a component of the microservices system, allowing users to securely input data through a user-friendly web interface. Operating on port 1207, it utilizes Express.js for web functionality. Authenticated users access the data entry form (EnterData.html) at the /EnterData route, where they can input various data types. The service stores this data in a MySQL database, creating the necessary database and table if not present. Data insertion is handled by the insertTemp function. Upon successful data entry, users are redirected to the "Show Results" service for analytics visualization at port 1209.

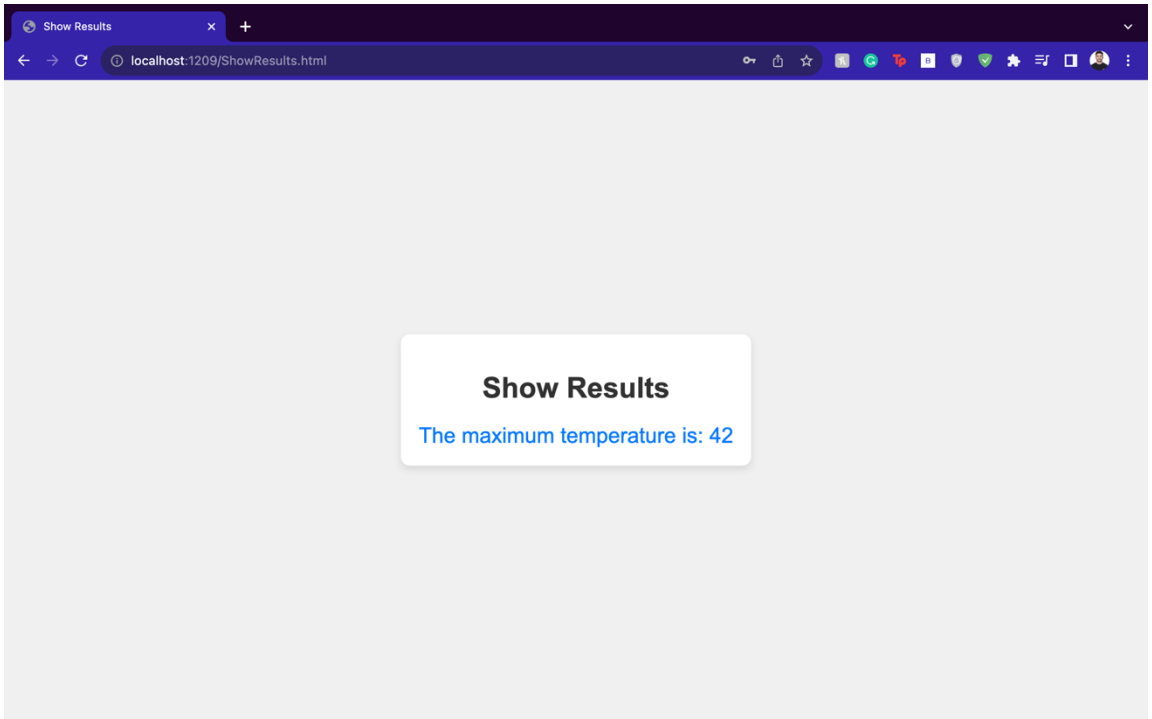




Show Results Web App:

The Show Results web app service, operating on port 1209, provides users with an interface to access and display analytical insights. It utilizes Express.js and MongoDB for efficient data handling. Authenticated users can access the /ShowResults route to view the latest recorded maximum temperature stored in the "max_temp" collection of the "maxDB" MongoDB database. The service ensures error handling and responsiveness by sending appropriate error responses if needed.





Authentication Service:

The Authentication"service, operating on port 1208, ensures secure user access and authentication within the microservices system. It employs Express.js to handle user requests and responses.

Using the /AutenticateEnt route, users are authenticated to access the Enter Data web app. If the provided email and password match the predefined credentials ("faresQ@atypon.com" and "fares!"), users are redirected to the Enter Data web app at http://localhost:1207/EnterData. Otherwise, they are redirected to the root route http://localhost:1207/.

Similarly, the /AutenticateRes route authenticates users to access the Show Results web app. If the users email and password match, they are redirected to http://localhost:1209/ShowResults.html. If not, they are redirected to the root route.

This service plays the role of securing access to the microservices and ensuring only authorized users can interact with the Enter Data and Show Results web apps.

Analytics Service:

The Analytics Service, residing in the 'analytics' container, drives dynamic analytical processes within the microservices framework. It interfaces with MySQL ('mysqldb') to retrieve maximum temperature data and then seamlessly connects with MongoDB ('mongo-db') to store insights. The '/Analytics' route orchestrates this process, fetching, saving, and presenting maximum temperature insights. Through this orchestrated coordination, the Analytics Service amplifies the architecture's data analytics potential, enhancing meaningful data extraction and storage capabilities.

MySQL DB:

The MySQL Database service functions as a key factor of the microservices ecosystem, providing a reliable and structured storage solution for various data. It operates on the MySQL platform and is used to store collected data from the Enter Data web app. Users' input data, in this case temperatures, are stored securely in the MySQL database.

The service, accessed through the 'mysqldb' container, is essential for persistent data storage and retrieval. It employs a predefined configuration, including the host, user, password, and port. This configuration ensures seamless communication between services within the system.

By creating a database named "Data" and a corresponding table named "Temps," the MySQL service eases organized data management. The database table structure supports efficient storage and retrieval of temperature values entered by users.

The MySQL DB service is a component that contributes to the integrity and structure of the microservices architecture, enabling secure, structured, and reliable data storage and access.

MongoDB:

The MongoDB Database service is a core element of the microservices framework, leveraging the MongoDB platform for dynamic and scalable data storage. Operating within the 'mongodb' container, it acts as a repository for analytical results from system services. By establishing the "maxDB" database and "max_temp" collection, MongoDB accommodates structured storage, especially for maximum temperature data. Configured through the 'mongoURL', the service facilitates seamless communication between microservices. Its flexibility supports efficient data retrieval, exemplified by the '/ShowResults' route in the Show Results web app, which fetches and displays the latest recorded maximum temperature. This adaptable service significantly contributes to the architecture's scalability and analytical capabilities.