

Project Continuance

SaaS Application for Fire Department 911 Risk Analysis

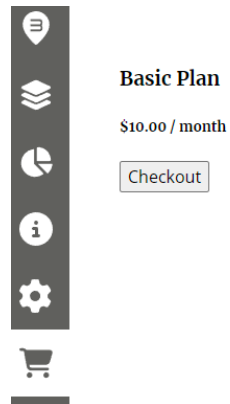
Masud Hussain, Samantha Jones, Amy Lee,
Anthony Kowalski, Zhiwei Ma, Marcos Valdez

Oregon State University
CS 467: Online Capstone Project – Fall 2023

Implemented Features

Control Plane

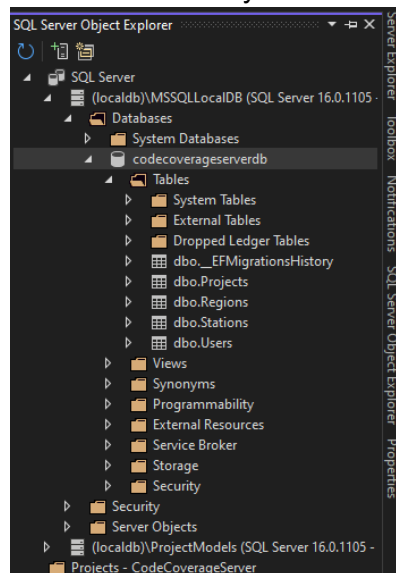
- User login with Auth0
- User checkout with Stripe



*Checkout redirects and interfaces
with external providers*

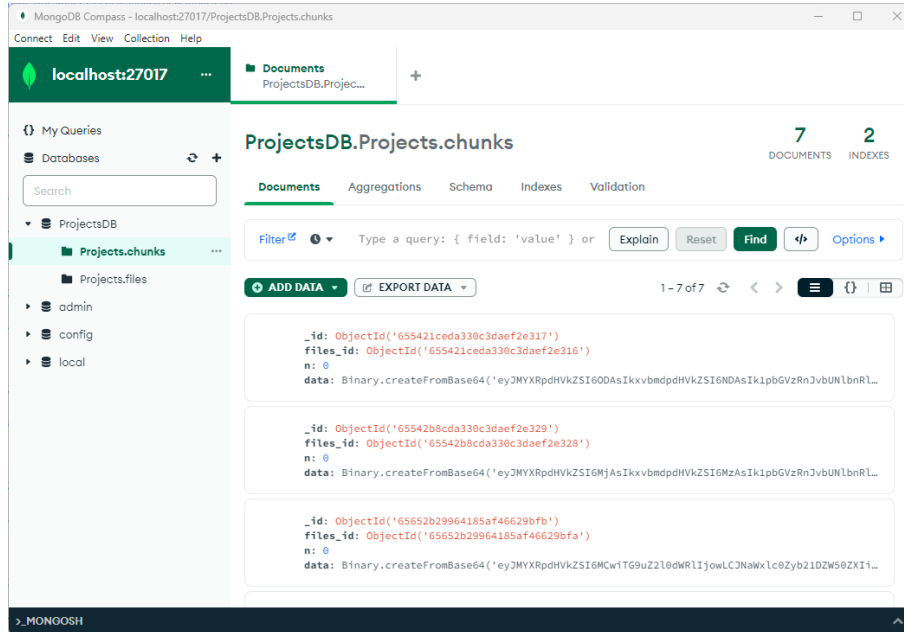
Database

- Relational database stores User data necessary for control plane



*Relational database (SQL Server)
entities shown in Visual Studio*

- GridFS implemented
 - Allows MongoDB to store and retrieve data that exceeds the BSON-document size limit of 16MB. It does so by converting the stored data to binary data and dividing it into chunks, which are then stored as separate documents. This resolved the primary problem with the previous release.



GridFS chunks shown in MongoDB Compass

- Database usage guides created for Windows/Linux and Mac

General

- Repository conflicts and merge plan documented
 - Codebases for **server-v2** and **server-summer23** are manually merged together in the migration branch on **server-v3**
 - Migration branch in **server-v3** has not been merged into the main branch due to conflicts with polyline calculations which has created a new task for the next team

Release Problems/Workarounds

Control Plane

- For Auth0, we will need to integrate it with the app such that the application will block/redirect the user to sign in/sign up unless they are logged in. Currently the user can use the application without signing in. This was a workaround to have the authentication working as an MVP.

Database

- Relational database is unused except for the User entity. CSVs provide source data; no workaround necessary for testing. See Next Phase Priorities.
- MongoDB database not integrated with the front end. Use **server-v2** repository, which does not store calculated nodes in a database, for frontend testing.

General

- Work split across two repositories. See Next Phase Priorities.
 - Test control plane, relational database and front end with **server-v2** repository
 - Test document database with **server-summer23** repository

Next Phase Priorities

Control Plane

- Future work might include extending payment methods like apple pay
- Monitoring and logging the performance of services that are offered
- Extend to other emergency services like ambulances, police stations, etc

Database

- Integrate server resource acquisition with database
 - SQL Server database exists but is empty
 - Resource (Location & Station) endpoints should read from SQL Server database rather than CSVs
- Integrate frontend with MongoDB
 - Internal usage for MongoDB is done in the CalculateCoverageResultsAsync() method of CoverageOSMRetriever.cs
 - There is an unresolved TODO comment within this method showing the point where the Project ID should be sent to the SQL database for storage

General

- Manually merge repositories **server-summer23** and **server-v2** so that the control plane, relational database, and document database are all implemented in the same server and properly interface with the frontend
- Research and find a way to rework the code in **server-summer23**, so polylines can be calculated with the updated code in **server-v2** which has GridFS and MongoDB implemented or simply add polyline calculations to **server-v2**.
- Deploy server and databases on Amazon cloud solutions