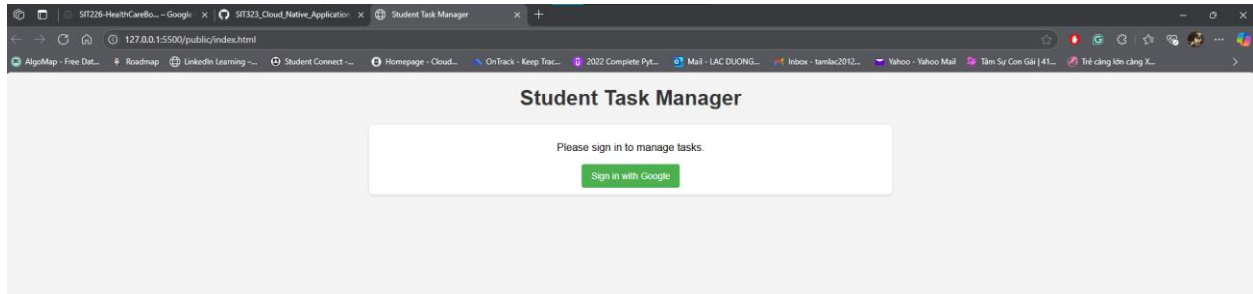


Video Link:

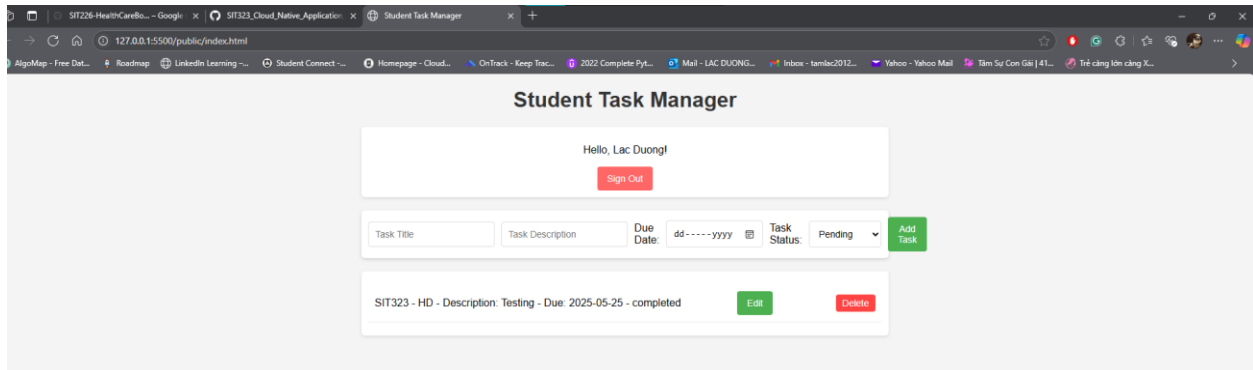
<https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=1d091864-a904-4d7a-b3a7-b2ea00e558b5>

Screenshots:

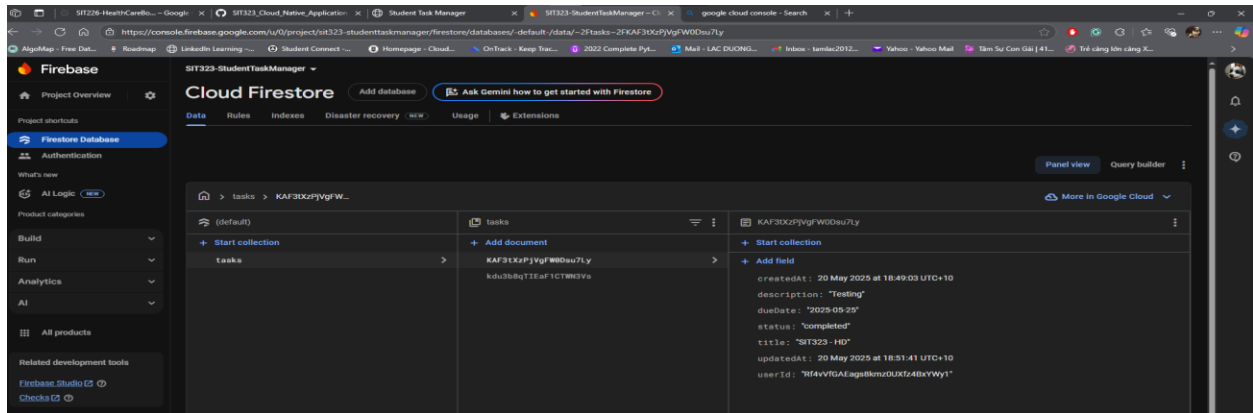
Sign in Page (currently not supporting signup feature, access via Google Gmail Account – Oauth protocol)



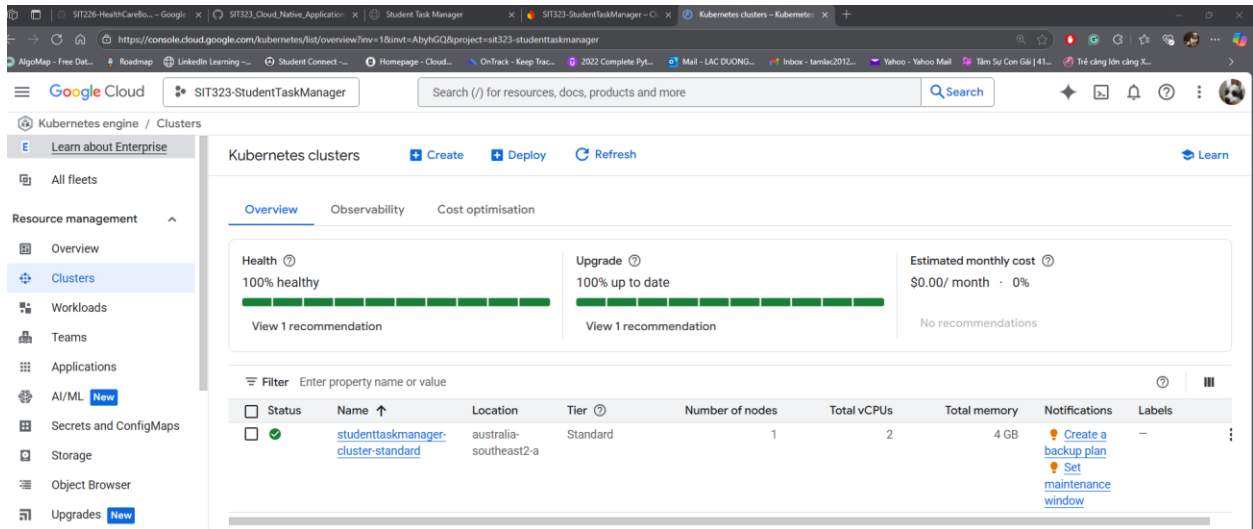
Main Dashboard of the website (allowing user to create a task through inputs, edit an existing task and delete that task)



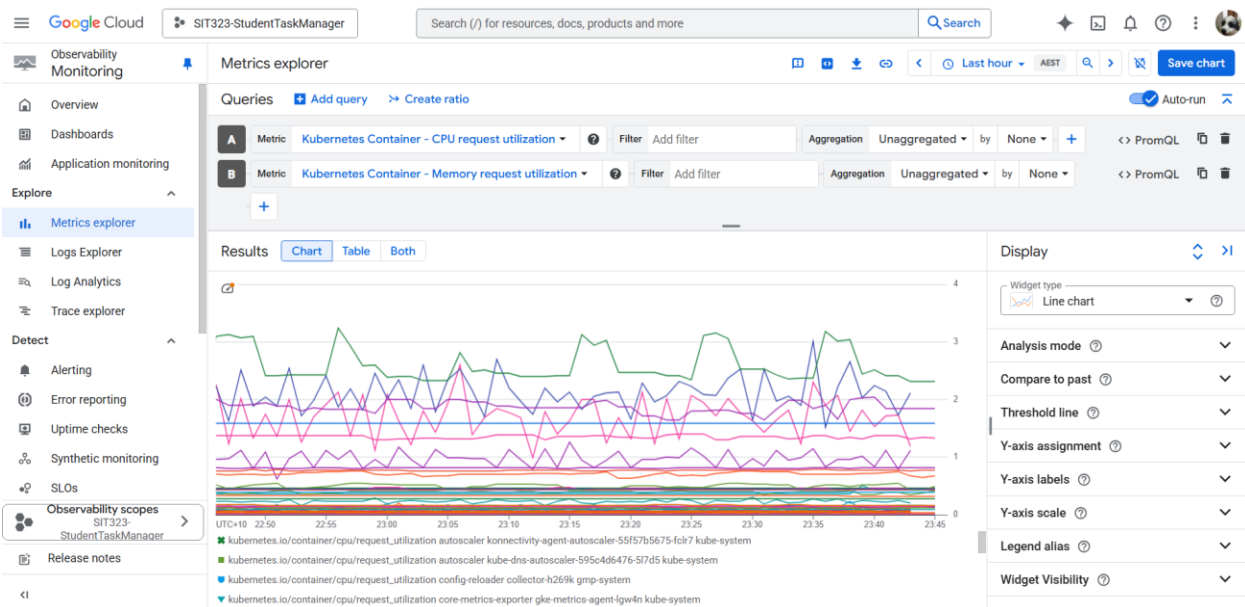
Cloud Firestore Database (tasks are added to this database, each task is stored as a document with information come from input and from the userId which is the login Gmail Account)



GKE Cluster & Node (currently using zonal instead of regional standard cluster, and 1 node is being used at the moment)



Monitoring with Metric Explorer services (CPU and Memory request utilization)



Worklog:

Week 1 (May 7-13)

May 7 - GCP Project Setup and Initial Configuration

- Created and configured GCP Project (SIT323-StudentTaskManager) for cloud-native development
- Enabled essential Google Cloud APIs:
- Cloud Firestore API for database functionality
- App Engine Admin API for potential future App Engine integration
- Cloud Build API for container image building
- Container Registry API for Docker image storage
- Kubernetes Engine API for container orchestration
- Established Cloud Firestore database in Native mode with appropriate region selection (australia-southeast2)
- Configured initial security rules in test mode to allow development and testing

May 8 - Backend Integration Setup

- Created dedicated service account (student-task-manager-dev) for development purposes
- Configured Firebase Admin SDK with appropriate permissions
- Implemented initial Firestore database connection in backend
- Started rewriting backend CRUD operations:
- Implemented task creation endpoint with Firestore integration
- Set up task retrieval functionality

May 9 - Backend Development Completion

- Completed implementation of all CRUD operations for Firestore:
- Finalized task update functionality
- Implemented task deletion with proper error handling
- Added data validation and error checking
- Conducted comprehensive testing of all API endpoints using Postman
- Verified data persistence and retrieval accuracy

May 10 - Containerization Setup

- Developed Dockerfile with Node.js 18 Alpine base image
- Configured container environment:
- Set up working directory
- Configured Node.js environment
- Set up port exposure
- Created comprehensive .dockerignore file to exclude unnecessary files
- Built initial Docker image locally
- Tested container functionality:
- Verified application startup
- Tested API endpoints within container

May 11 - Container Registry Integration

- Configured Docker client for Google Container Registry access
- Set up authentication for GCR
- Tagged Docker image with appropriate versioning
- Pushed image to Google Container Registry
- Verified image availability and accessibility

Week 2 (May 14-20)

May 14 - Kubernetes Cluster Setup

- Created GKE cluster in australia-southeast2-a region
- Configured cluster specifications:
- Set up node configuration
- Configured network settings
- Established security parameters
- Set up kubectl for cluster management
- Verified cluster connectivity and configuration

May 15 - Kubernetes Configuration

- Developed Kubernetes deployment YAML
- Configured pod specifications
- Set up container configurations
- Implemented resource limits
- Created LoadBalancer service configuration
- Implemented Workload Identity for secure GCP service access
- Applied initial Kubernetes configurations

May 16 - Application Deployment

- Deployed application to GKE cluster
- Monitored deployment progress and pod health
- Verified service accessibility through LoadBalancer
- Conducted initial load testing
- Troubleshoot deployment issues

May 17 - Frontend Integration

- Updated frontend code to connect with GKE backend
- Implemented CORS configuration
- Resolved connection and communication issues
- Conducted frontend testing:
- Verified task creation
- Tested task updates
- Validated task deletion

May 18 - Authentication Implementation

- Set up Firebase Authentication project
- Implemented user interface components:
- Created login form
- Developed signup interface
- Added user profile management
- Integrated Firebase SDK with frontend application
- Implemented authentication state management
- Added user session handling

May 19 - Security Implementation

- Secured backend endpoints with Firebase authentication
- Implemented user-specific task management:
- Added user ownership to tasks
- Implemented task access control
- Developed token verification middleware
- Updated task operations to include user context

May 20 - Final Testing and Documentation

- Conducted comprehensive end-to-end testing:
- Tested user authentication flow
- Verified task management functionality
- Validated security measures
- Resolved final authentication and authorization issues
- Prepared final project report

PowerPoint Slides Link: (I attached this Slides within the folder of task 10.2HD)

https://github.com/HaydenDuong/SIT323_Cloud_Native_Application_Development/tree/main/Student%20Task%20Manager/documents