

About My Home Town: Project Plan

CSC478: Software Engineering Capstone

Audrey Yates, William Morgan, Steven Hogue
Group 11

May 2, 2024

Contents

1	Scope Statement	2
1.1	Overview	2
1.2	Platform	2
1.3	Infrastructure	2
1.4	Integrations	3
1.5	Programming Languages	3
1.6	Timeline	3
2	Organization Chart	3
3	Gantt Chart	4
4	Tools and Standards	4
4.1	Development	4
4.1.1	Tools	4
4.1.2	Applications	4
4.2	Standards	5
5	Configuration Management	5
5.1	GitHub	5

1 Scope Statement

1.1 Overview

Our website will be viewable at <https://www.aboutmyhometown.com>.

We will construct a cross-platform web application that allows users to check on a variety of data points for their locale. This data may include, but is not limited to, gas prices, current and forecast weather, restaurant/food prices, public facilities, local businesses, and possibly crime information. The goal is to compile as much data from public or low-cost API's into a single interface, adding statistical analysis, and overlaying it into a mapping application (such as Google Maps). As a long-term goal we would like to interpret and read news articles relating to an area of interest, and pin the location in which they took place on the map.

1.2 Platform

This interface will be available through any modern, commonly used web browser. As the application is a web application, an internet connection and a modern browser is all that is required. The goal is to allow it to look good and work on a wide range of OS's and devices.

1.3 Infrastructure

The tech stack for this project will include Angular for front end development, a GitHub repository for project storage and CI/CD, online JIRA (team.atlassian.com) for ticket tracking, with the primary infrastructure being hosted in AWS.

Project Management:

- **GitHub:** Project CI/CD pipelines and code versioning
- **Jira:** Team, work, and project tracking

Tech stack:

- **Angular:** Front-end and interface development
- **Bootstrap:** Easy-to-use and cross platform formatting engine
- **AWS:** Hosting and server resources
 - **S3:** Static website host
 - **Lambda:** Server-side processing, done primarily in Python
 - **Database:** MySQL and DynamoDB, for handling long-term storage of users and fast-changing API data respectively
 - **API Gateway:** Handle traffic and routing management
 - **Certificate Manager:** Ensure up-to-date SSL certificate
 - **Route 53:** DNS server
 - **CloudFront:** High speed performant content delivery network (CDN)

- **CloudFormation:** Used to configure the cloud resources from managed templates

1.4 Integrations

Our plan will need to integrate with a number of other web services, including lots of data gathering APIs. Other APIs will include a map API such as OpenStreetMap or Google Maps, pending available functionality. User registration and login will be required, with each user specifying their location (see Timeline). Parts of the authentication will be stored both in browser local storage and the database.

1.5 Programming Languages

This project will use TypeScript (Angular), HTML, and CSS for the frontend user interface, with Python being used in Lambda functions as the server-side processor. It will also include SQL for database management, and AWS specific CloudFormation templates for infrastructure management.

1.6 Timeline

The initial focus will be on the Springfield area until the product is refined. Once that is satisfactory, if time permits, we would like to expand the scope to include other areas in Illinois or further. The goal is to make the site work dynamically based on the user's location, though that could end up being too complicated as available data reaches its limits.

2 Organization Chart

Frontend Design: Audrey Yates

Design the user interface, including the layout, color scheme, and overall look of the application.

API Integrations: Bill Morgan

Integrate the various APIs that will be used to gather data for the application.

Architecture: Steven Hogue

Implement the overall architecture of the application, including the database, user authentication, and CI/CD.

3 Gantt Chart

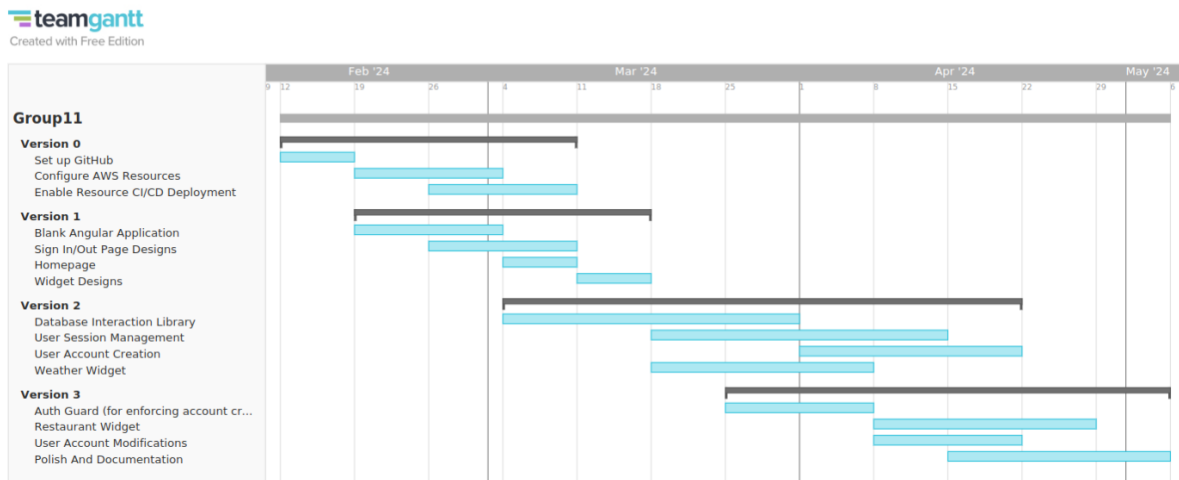


Figure 1: Gantt Chart

4 Tools and Standards

4.1 Development

4.1.1 Tools

- **NodeJS:** 20.0.0+
 - The core for building the application
- **Angular:** 14.3.0+
 - The architecture that the application will be built on
- **Python:** 3.11.0+
 - Needed for some of the Lambda functionality
- **Docker:** 26.1.0+
 - Needed in order to run the application locally, primarily for use with the AWS SAM CLI in deploying local copies of the Lambda functions
- **AWS SAM CLI:** 1.113.0+
 - Preferred command line utility for all deployments to AWS, and local testing of Lambda functions

4.1.2 Applications

- **VS Code**
 - The code editor and IDE of choice for our team

- **Google Chrome or Mozilla Firefox**
 - These are the browsers that the application will be tested on by default

4.2 Standards

- **Python PEP 8 Style Guide**
- **Agile Development**

5 Configuration Management

5.1 GitHub

The project will be stored in a GitHub organization, with the main branch being the primary branch for the project. All changes will be made in feature branches, and merged into the master branch via pull requests. The project will be built and deployed via GitHub Actions.

For release management, the project will use GitHub releases to track the progress of the project, and to mark major milestones. This will be done simultaneously using tags and releases.

GitHub Organization: <https://github.com/AboutMyHT>

Angular Application Repository: <https://github.com/AboutMyHT/angular-app>

AWS Resources Repository: <https://github.com/AboutMyHT/aws-resources>