

Deployment Report

Nutritional Insights

Project 2

Submitted by:
. , Kaur, Komalpreet
Ala, Anne Marie
Cona, Rhailyn Jane
Software Development, SADT

November 03, 2025

Executive Summary

This report documents the successful deployment of the Nutritional Insights web application using Microsoft Azure Static Web Apps. The application was deployed with automated continuous integration through GitHub, enabling seamless updates and reliable cloud hosting. The deployment process was completed using Azure's Free hosting plan under an Azure for Students subscription.

1. Introduction

1.1 Project Overview

The Nutritional Insights Project 2 is a web-based dashboard application designed to visualize nutritional data through interactive charts and statistical representations. The application provides insights into dietary patterns, macronutrient distributions, and protein content across various recipe types.

1.2 Deployment Objective

The primary objective was to deploy the web application to a cloud platform that offers:

- Reliable and scalable hosting
- Continuous deployment capabilities
- Cost-effective solutions for academic projects
- Integration with version control systems

1.3 Technology Stack

- **Cloud Platform:** Microsoft Azure Static Web Apps
- **Version Control:** GitHub
- **Deployment Method:** GitHub Actions (CI/CD)
- **Subscription Type:** Azure for Students
- **Hosting Plan:** Free Tier

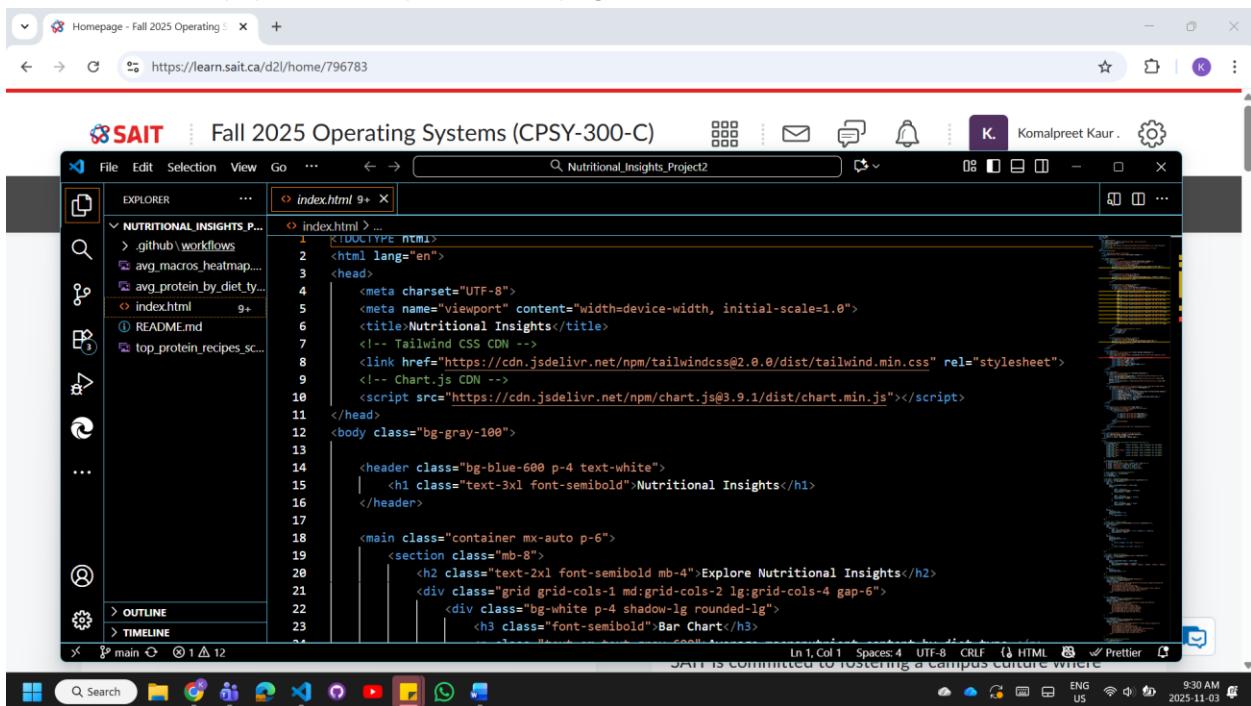
2. Deployment Methodology

2.1 Phase 1: Project Preparation

The initial phase involved organizing and optimizing project files for cloud deployment. Key activities included:

File Structure Optimization

- Renamed the main HTML file to index.html to serve as the application's entry point
- Organized visualization assets including:
 - avg_macros_heatmap.png
 - avg_protein_by_diet_type.png
 - top_protein_recipes_scatter.png



- Removed unnecessary backend folders and unused data files to reduce deployment package size
- Ensured all file paths were relative and compatible with static hosting requirements

Quality Assurance

- Verified all hyperlinks and resource references
- Tested the application locally before deployment
- Confirmed all visualizations rendered correctly

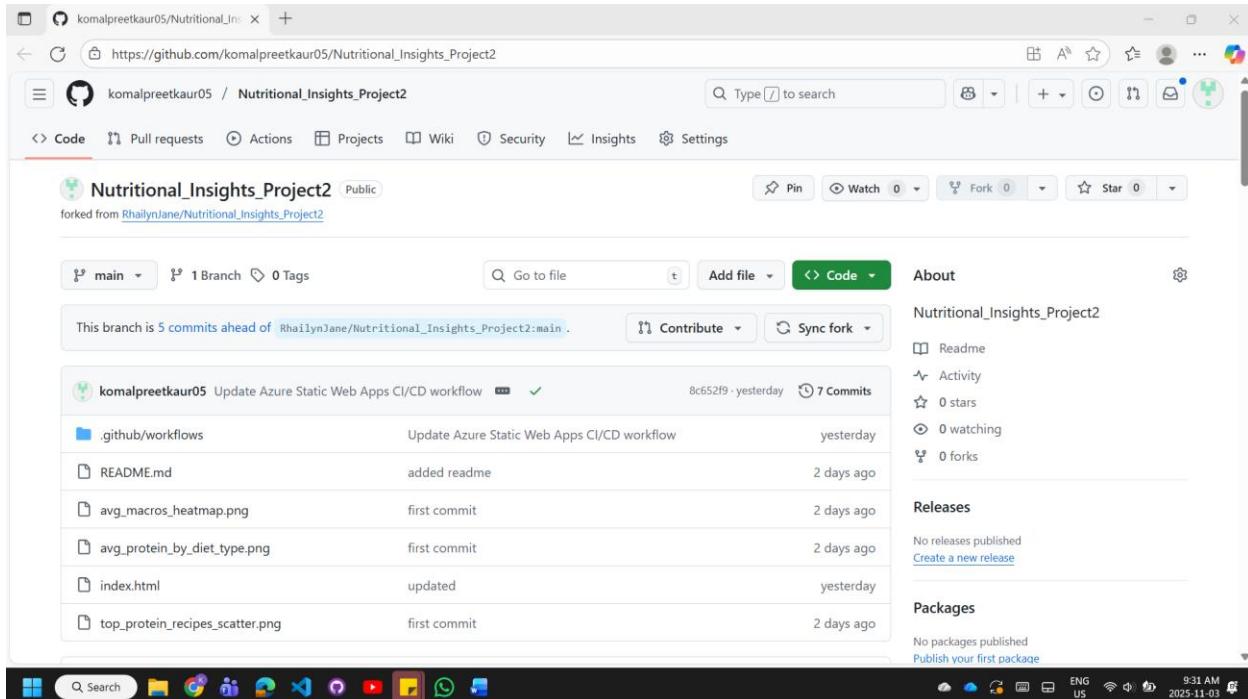
2.2 Phase 2: GitHub Repository Setup

Version control was established through GitHub to enable continuous deployment capabilities.

Repository Configuration

- Created a new repository named Nutritional_Insights_Project2
- Placed all project files in the root directory for simplified Azure integration
- Committed files to the main branch to establish the deployment source
- Configured repository settings to allow Azure integration

This setup ensured that all future updates could be automatically synchronized with the deployed application through GitHub Actions workflows.



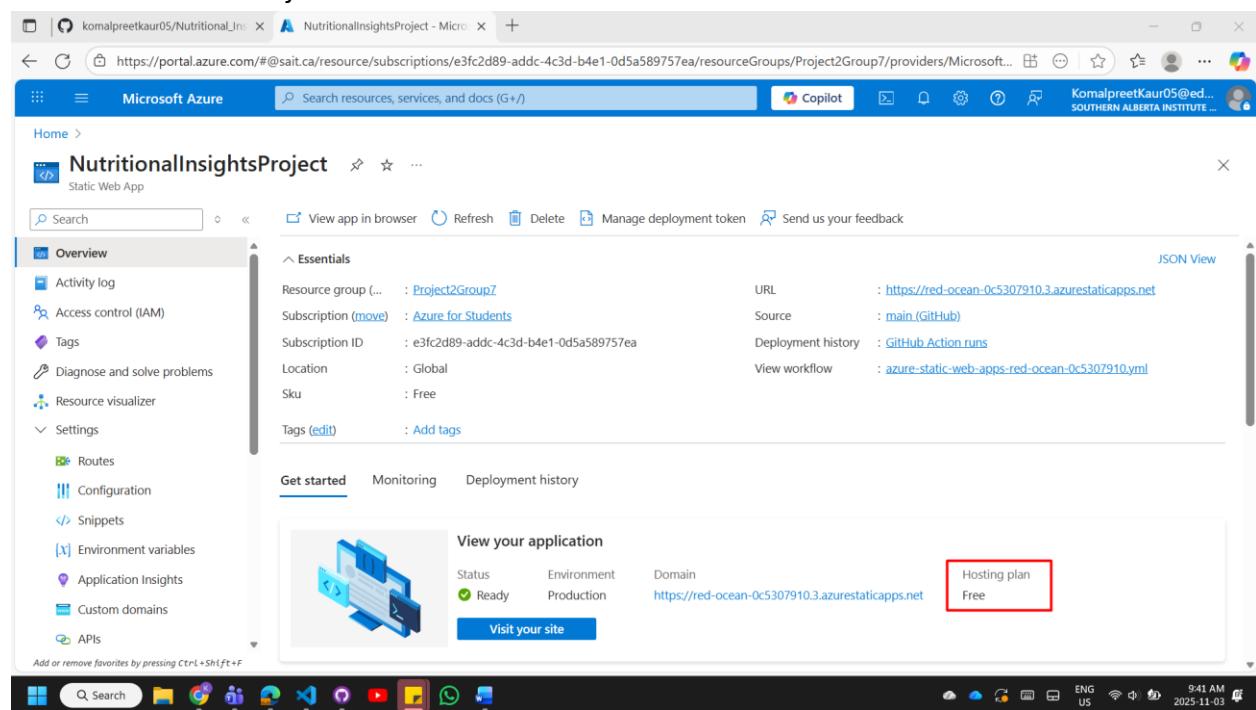
2.3 Phase 3: Azure Static Web App Creation

A new Azure resource was provisioned specifically for hosting the web application.

Resource Configuration

- **Subscription:** Azure for Students
- **Resource Group:** Project2Group7 (newly created)
- **Application Name:** NutritionalInsightsProject
- **Region:** Central US
- **Hosting Plan:** Free (Standard Tier 0)

The Free hosting plan was selected to minimize costs while providing adequate resources for an academic project. Central US was chosen as the deployment region to ensure reasonable latency and service availability.



The screenshot shows the Microsoft Azure portal interface for managing a static web app named "NutritionalInsightsProject". The left sidebar contains navigation links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings (Routes, Configuration, Snippets, Environment variables, Application Insights, Custom domains, APIs), and Get started. The main content area displays the "Essentials" section with details such as Resource group (Project2Group7), Subscription (Azure for Students), Subscription ID (e3fc2d89-addc-4c3d-b4e1-0d5a589757ea), Location (Global), Sku (Free), and Tags (Add tags). Below this, there's a "View your application" card showing Status (Ready), Environment (Production), Domain (https://red-ocean-0c5307910.3.azurestaticapps.net), and a "Visit your site" button. A red box highlights the "Hosting plan Free" text in the bottom right corner of the application view card.

2.4 Phase 4: GitHub Integration

The Azure Static Web App was connected to the GitHub repository to enable automated deployments.

Integration Settings

- **GitHub Account:** komalpreetkaur05
- **Repository:** Nutritional_Insights_Project2
- **Branch:** main
- **Build Preset:** HTML
- **App Location:** / (root directory)
- **API Location:** (blank - no backend API)
- **Output Location:** / (root directory)

This configuration established a direct link between the GitHub repository and Azure, enabling continuous integration and continuous deployment (CI/CD). Any commits pushed to the main branch would automatically trigger a redeployment of the application.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Azure logo, account information (komalpreetkaur05/Nutritional_Insights_Project2), and a search bar. Below the header, the URL https://portal.azure.com/#@sait.ca/resource/subscriptions/e3fc2d89-addc-4c3d-b4e1-0d5a589757ea/resourceGroups/Project2Group7/providers/Microsoft... is visible. The main content area displays the 'NutritionalInsightsProject | Configuration' page for a 'Static Web App'. On the left, a sidebar menu lists various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, Routes, and Configuration. The 'Configuration' option is currently selected. The main pane shows configuration details: 'Enable configuration file changes' (checkbox checked), 'Deployment configuration' (describing GitHub Identity or Azure Access Token deployment), 'Deployment authorization policy' (radio button selected for 'GitHub'), 'Signed in as' (komalpreetkaur05), and 'Build Details' (App location set to '/'). At the bottom of the configuration pane are 'Apply' and 'Discard' buttons. The status bar at the bottom right shows the date and time (9:42 AM 2025-11-03) and system icons.

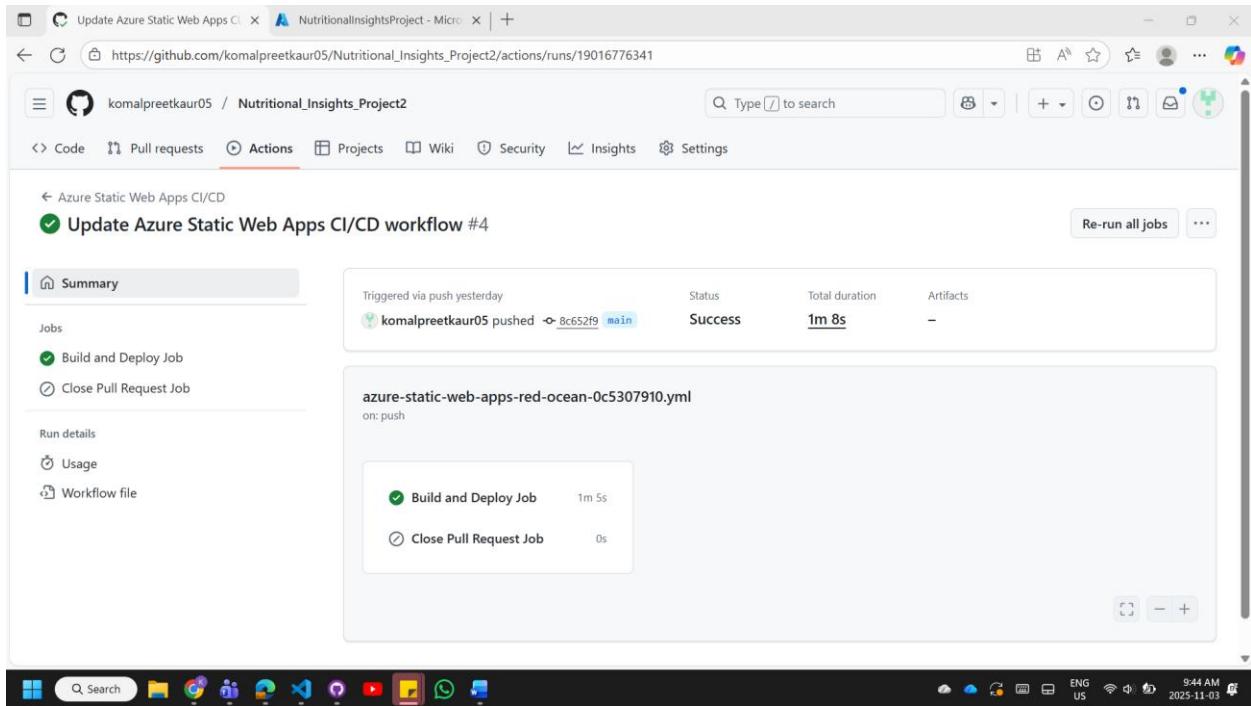
2.5 Phase 5: GitHub Actions Workflow Configuration

Azure automatically generated a GitHub Actions workflow file to manage the build and deployment process.

Workflow Configuration

- **Location:** .github/workflows/ directory in the repository
- **Format:** YAML configuration file
- **Build Setting:** skip_app_build: true

The workflow was configured to skip unnecessary build steps since the project consists solely of static files. This optimization reduced deployment time and simplified the CI/CD pipeline. The workflow executes automatically upon each commit to the main branch, ensuring the deployed application remains synchronized with the latest code changes.



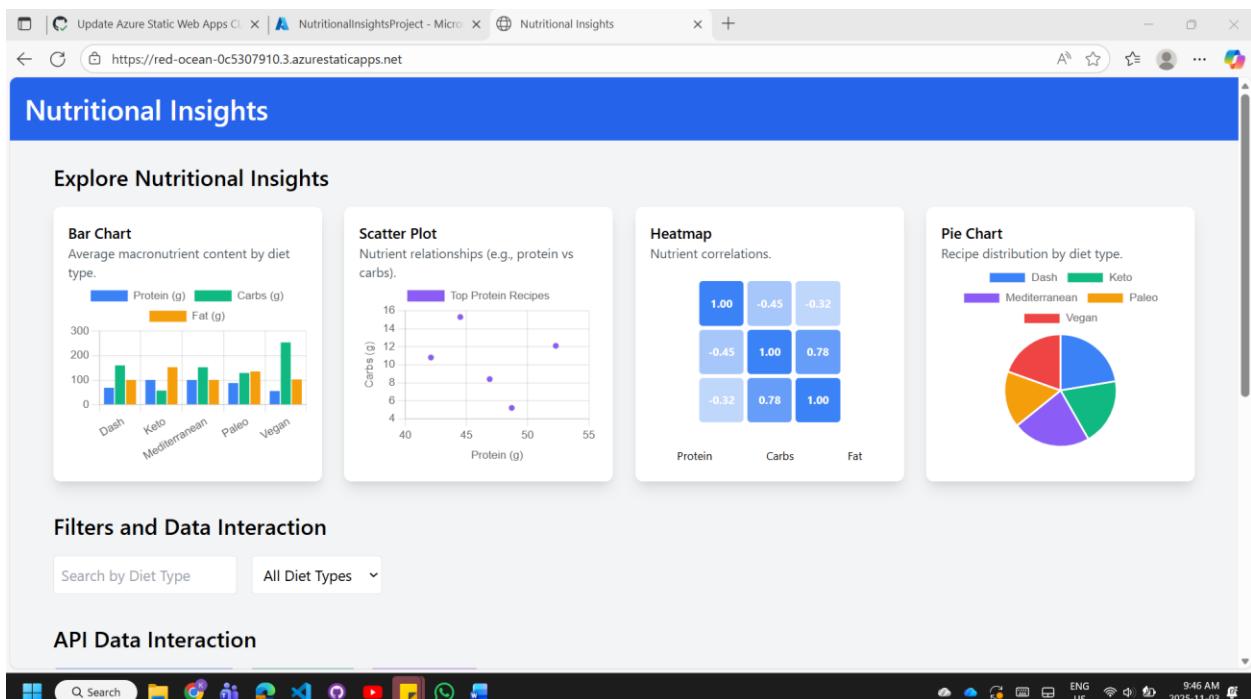
2.6 Phase 6: Deployment Verification

Post-deployment validation was performed to ensure successful hosting and functionality.

Verification Activities

- Accessed the application via the assigned Azure URL
- Confirmed all pages loaded correctly
- Verified that all visualizations rendered properly
- Tested interactive elements and navigation
- Validated continuous deployment by confirming GitHub Actions workflow execution

All verification tests passed successfully, confirming that the application was properly deployed and accessible to end users.



3. Deployment Architecture

3.1 Infrastructure Overview

The deployment utilizes Azure's globally distributed content delivery network (CDN) to serve static content efficiently. The architecture includes:

- **Front-end Hosting:** Static files served through Azure Static Web Apps
- **Content Delivery:** Automatic CDN distribution for optimized performance
- **SSL/TLS:** Automatic HTTPS encryption for secure connections
- **Version Control Integration:** GitHub as the source repository
- **Automation:** GitHub Actions for CI/CD pipeline

3.2 Continuous Deployment Pipeline

The CI/CD pipeline operates as follows:

1. Developer pushes commits to the main branch on GitHub
2. GitHub Actions workflow is automatically triggered
3. Workflow validates and packages the static files
4. Azure Static Web Apps pulls the updated content
5. Application is automatically redeployed with zero downtime
6. New version becomes immediately available at the Azure URL

This automated pipeline ensures rapid deployment of updates while maintaining application availability.

Conclusion

The Nutritional Insights Project 2 has been successfully deployed to Microsoft Azure Static Web Apps, establishing a robust, cloud-hosted web application with continuous deployment capabilities. The integration with GitHub ensures that future enhancements can be deployed automatically, creating a reliable and efficient update mechanism.

The deployment process demonstrated the effectiveness of modern cloud platforms for hosting static web applications, particularly for academic projects requiring professional presentation without significant infrastructure investment. The resulting application is publicly accessible, secure, and maintainable through standard development workflows.