Secure Software Development – Group Deployment Report

**Project: Dizzy's Disease**Generated: 26 October 2025, 00:25 UTC  
Prepared by: SSD Deployment Team

# 1. Assignment Purpose

This report documents the deployment of the Dizzy's Disease application to the shared staging environment required for the Secure Software Development course. It also records evidence that the deployed system is operational and that reviewers have read-only access to the complete source code.

# 2. Deployment Summary

|  |  |
| --- | --- |
| Staging URL | https://thankful-wave-09a5b000f.3.azurestaticapps.net/ |
| Hosting Platform | Azure Static Web Apps – Free tier (East US 2) |
| Backend | Azure App Service (https://dizzy-api-app.azurewebsites.net/) |
| Database | Azure PostgreSQL Flexible Server – server dizzy-db-57a0, database 'dizzy' |
| Client Build | Godot 4.4 HTML export located at capstone/build/web/ |
| CI/CD | GitHub Actions workflow .github/workflows/deploy-static-web-app.yml (action=upload) |

# 3. Rubric Compliance

Evidence aligned to the marking rubric is summarised below.

|  |  |  |
| --- | --- | --- |
| Criterion | Evidence | Comments |
| Operation of Deployed System | Staging build online at https://thankful-wave-09a5b000f.3.azurestaticapps.net/. Registration, login, Continue/New Game, inventory and offline mode tested after latest deployment. | Behaviour verified in browser and via Playwright console capture. |
| Access to Buildable Source | Classroom repository: https://github.com/Steve-at-Mohawk-College/capstone-project-Jinphinity Personal mirror: https://github.com/Jinphinity/capstone | Both repositories contain the Godot project, FastAPI backend, and deployment workflow. |
| Access Instructions for Reviewers | Detailed tester workflow included in Section 4 (login, health check, offline mode, API inspection). | Supports ethical hacking review. |
| Document Format | This formal DOCX report plus JSON/TXT evidence in docs/ssd/deliverables/ | Organised with sections, tables, and appendices. |

# 4. Tester Instructions

Reviewers should complete the following steps when evaluating the deployment:

1. Navigate to https://thankful-wave-09a5b000f.3.azurestaticapps.net/ (Chrome or Edge recommended).
2. Register a new account or log in with an existing tester account. Passwords are hashed with bcrypt on the backend.
3. After login, confirm that the Continue/New Game menu appears and that inventory/market interactions function.
4. Select 'Skip Login' to verify offline mode operation.
5. Open browser developer tools and confirm API traffic is sent to https://dizzy-api-app.azurewebsites.net/ (e.g., /auth/login, /market/prices).
6. Check API health at https://dizzy-api-app.azurewebsites.net//health (expect JSON response with status='healthy').
7. Optional: execute the redeploy command shown in Section 6 using the exported Web build.

# 5. Supporting Evidence

Artifacts supplied with this submission:

• SSD\_Group\_Deployment\_Report.docx – This document.

• api\_health.json – Captured response from the live API health endpoint.

• staging\_info.txt – Quick reference containing staging URL, GitHub Actions runs, and curl commands.

# 6. Operational Notes

Manual redeployment (requires Azure CLI and SWA CLI):

swa deploy ./capstone/build/web --app-name dizzy-disease-swa --resource-group DizzySWA-RG --subscription-id eedd13b0-fd70-45bc-99a0-7d1be4d779ec --env production

Backend container image: dizzyacr57a0.azurecr.io/dizzy-api:latest

Database connection string: postgresql://dbadmin@dizzy-db-57a0/dizzy?sslmode=require

# Appendix A – GitHub Actions Runs

https://github.com/Jinphinity/capstone/actions/runs/18809305400

https://github.com/Steve-at-Mohawk-College/capstone-project-Jinphinity/actions/runs/18809048337