Video Demo:

https://deakin.au.panopto.com/Panopto/Pages/Viewer.aspx?id=1cee5188-00b0-4d7d-bc34-b2ed00c77d80

Outline of HealthcareBooking_Final_Artifacts.zip Contents and Relevance

The HealthcareBooking_Final_Artifacts.zip file contains all source code, configuration files, database scripts, and supporting evidence demonstrating the successful development, deployment, and advanced functionality of the Healthcare Appointment Booking System.

These artifacts collectively showcase the achievement of project objectives and Unit Learning Outcomes at an advanced level.

1/ Application Source Code:

- **app.py**: The core Flask application file. Contains the Python code for all web routes (frontend rendering, API endpoints for booking and retrieving appointments), database connection logic using psycopg2, and interaction with the PostgreSQL database. *Demonstrates backend development and database integration*.
- **templates/book.html**: The HTML template for the user-facing appointment booking interface. *Showcases frontend design and user interaction elements*.
- **static/style.css**: CSS file providing styling for the frontend. *Contributes to user experience and interface presentation*.

2/ Containerization Configuration:

- **deployment/Dockerfile**: Instructions to build the Docker image for the Flask application. Specifies the base image, dependencies, code copying, port exposure, and runtime command. *Demonstrates application packaging for portability and reproducible deployments, a key aspect of ULO2 and ULO5*.
- **deployment/requirements.txt**: Lists Python dependencies (Flask, psycopg2-binary) for the application. *Ensures consistent environment setup*.

3/ Kubernetes Configuration Files (Located in deployment/ directory):

- **flask-deployment.yaml**: Defines the Kubernetes Deployment for the Flask application. Manages stateless application pods, specifies the Docker image (e.g., hayden2310/healthcarebooking_app:v2), replica count, port configuration, environment variables, resource requests/limits, and readiness/liveness probes. *Key for demonstrating ULO1 (resource representation), ULO2 (deploying and managing applications), and ULO5 (optimizing deployment with health checks and resource management)*.
- **flask-service.yaml**: Defines the Kubernetes Service (ClusterIP) that provides a stable internal network endpoint for accessing the Flask application pods. *Illustrates ULO2 in managing network services for applications*.
- **flask-hpa.yaml**: Defines the HorizontalPodAutoscaler for the Flask application. Configured to automatically scale pods based on CPU utilization. *Crucial for demonstrating advanced application of ULO5 (optimizing management through autoscaling) and ULO1 (dynamic resource management)*.
- **postgres-statefulset.yaml**: Defines the Kubernetes StatefulSet for deploying the PostgreSQL database. Manages stateful application pods, ensures stable network identity, configures persistent storage via volumeClaimTemplates, and sets environment variables for database initialization (including PGDATA for correct volume initialization). *Showcases advanced understanding of ULO2 (deploying stateful services) and ULO5 (reliable data management)*.
- **postgres-service.yaml**: Defines the headless Kubernetes Service for the PostgreSQL StatefulSet, providing stable network identifiers for database pods. *Essential for enabling reliable database connectivity from the Flask application within Kubernetes (ULO2)*.

4/ Database Scripts:

- **db_setup.sql**: Contains the SQL CREATE TABLE statement for the appointments table, defining the initial database schema. *Fundamental for application data structure (ULO4 - design elements)*.
- **optimization.sql**: Contains the SQL CREATE INDEX statement for adding an index to the appointments table on the appointment_time column. *Demonstrates a database optimization technique (ULO5 - performance optimization)*.

5/ Documents

- Evidences.docx: documented coding journey, challenges / issues encountered and way to overcame along with pictures obtained as I progressed through this project.