13. Deployment of Flask Web Application with Embedded Tableau Dashboard

13.1 Overview

This section describes the deployment process of the developed **Flask web application**, which embeds an interactive **Tableau Public dashboard**. The application presents insights from *A College Food Choices Case Study* and has been hosted using **Render.com**, a cloud platform well-suited for deploying Python web services

13.2 Hosting Platform

- **Platform:** Render.com
- **URL:** https://render.com
- **Purpose:** To host the Flask application on a publicly accessible URL without requiring complex DevOps setup.
- **Reason for Selection:** Render provides free-tier services, native support for Python/Flask apps, easy GitHub integration, and automatic builds.

13.3 Project Structure

The Flask application was structured as follows:

```
/flask
├── app.py              # Main Flask application logic
├── requirements.txt      # Project dependencies for deployment
├── Procfile            # Specifies how to run the app using Gunicorn
├── templates/
│   └── index.html       # HTML template embedding the Tableau dashboard
├── static/             # Optional folder for CSS/JS or static assets
```

13.4 Key Configuration Files

13.4.1 requirements.txt

Defines the Python dependencies required by the project. This file ensures Render installs the correct packages during deployment.

```
Flask==2.3.2
gunicorn==21.2.0
```

13.4.2 Procfile

Instructs the Render platform to launch the Flask app using Gunicorn (a production-ready WSGI server).

```
web: gunicorn app:app
```

Note: app:app refers to the filename (app.py) and the Flask instance (app).

13.5 Deployment Process

The following steps were followed to deploy the application:

1. **Repository Setup**
   - The Flask project was uploaded to a public GitHub repository:
     🔗 https://github.com/Rajesh26013/flask
2. **Connecting to Render**
   - Logged into Render using GitHub credentials.
   - Selected "New Web Service" and connected the repository.
3. **Configuration Settings**
   - **Build Command:** pip install -r requirements.txt
   - **Start Command:** gunicorn app:app
   - **Runtime Environment:** Python 3 (auto-detected)
4. **Automatic Build & Deployment**
   - Render cloned the repository, installed dependencies, and launched the Flask app.
   - A public URL was generated for accessing the live application.

13.6 Issue Encountered and Resolution

During the initial deployment, the following error occurred:

```
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
```

Cause: The requirements.txt file was missing from the repository.

Resolution:
The file was manually created with the appropriate dependencies, committed, and pushed to the GitHub repository. After re-triggering the deployment, the issue was resolved and the application deployed successfully.

13.13 Final Result

Once deployed, the Flask application successfully rendered the embedded Tableau dashboard, allowing users to interactively explore the food and nutrition data collected as part of the case study.

deployed URL: Health and Nutrition Dashboard

13.8 Conclusion

The deployment process illustrates a streamlined approach to hosting data visualizations through Flask and Tableau using Render. This solution enables the delivery of dynamic dashboards to end-users via a lightweight, scalable, and cost-effective platform.