

Deployment Instructions and Considerations

The **Recipe Hub** application was deployed on **Render.com** using a connected GitHub repository. The deployment process was designed to ensure that the hosted application runs reliably in a production environment while closely mirroring the local development setup.

Deployment Steps

1. Repository Setup

- The source code was pushed to GitHub:
 <https://github.com/Tanvi-Bagwe/recipe>
 - Render was linked to the repository to enable **continuous deployment**.
-

2. Render Configuration

- Deployment was managed through a **render.yml** file for **reproducibility and automation**.
- Key configurations defined in the YAML:
 - **Runtime:** Python
 - **Build Command:** pip install -r requirements.txt
 - **Start Command:** gunicorn app:app
 - **Region:** Oregon (default free tier region)
 - **Plan:** Free tier (with limitations like auto-sleep after inactivity).
 - **Domains:**
 - recipehub.pro - www.recipehub.pro

3. Service Deployment

- From the **Render Dashboard** → **Blueprints** → **New Blueprint Instance**, the GitHub repo was connected.
 - Render automatically detected the **render.yml** file and built the project.
 - The service was deployed at the default Render URL and later mapped to a custom domain.
-

4. Custom Domain Configuration

- A custom domain was purchased from **GoDaddy** and pointed to Render:
 - **A Record** → **216.24.57.1** for @ (root domain).
 - **CNAME Record** → **recipe-hub-1xes.onrender.com** for www.
 - Domain ownership was verified, and **SSL certificates** were automatically provisioned by Render for secure HTTPS access.
-

5. Auto-Deployments

- Auto-deploy was enabled (autoDeployTrigger: commit in YAML).
- Every new commit to the GitHub repository automatically triggers a **build and deployment** on Render.

Deployment Considerations

- **Free Plan Limitations:**
 - On the free Render plan, the web service sleeps after 15 minutes of inactivity.
 - To mitigate this, a **Render Scheduler** pings the site every 5 minutes using Python's requests module to keep the service active.
- **Environment Consistency:**
 - All required dependencies (Flask, Flasgger, Gunicorn, etc.) were included in requirements.txt.
 - Verified that gunicorn app:app correctly runs the Flask app in production mode.
- **Post-Deployment Verification:**
 - All key pages (Home, Recipes Grid, Recipe Details, About Us, Testimonials) were tested for correct rendering.
 - API endpoints (e.g., /testimonial/all/, /testimonial/add/) were validated.
 - Swagger UI (/apidocs) was confirmed functional for documenting APIs.
 - Styling, JSON data persistence, and interactivity were checked to ensure the deployed app matched the local development experience.
- Render.yml

```
services:  
  - type: web  
    name: recipe-hub  
    runtime: python  
    repo: https://github.com/Tanvi-Bagwe/recipe  
    plan: free  
    region: oregon  
    buildCommand: pip install -r requirements.txt  
    startCommand: gunicorn app:app  
    domains:  
      - recipehub.pro  
      - www.recipehub.pro  
    autoDeploy: true
```