DineWise Deployment Guide

Table of Contents

Local Development Setup

Prerequisites

- Node.js 16+ and npm
- Python 3.8+
- Git
- Firebase account

Frontend Setup (React)

1. Clone the repository:

```
git clone [your-repo-url]
cd dinewise
```

2. Install dependencies:

```
npm install
```

3. Create a .env file with your Firebase configuration:

```
VITE_FIREBASE_API_KEY=your_api_key
VITE_FIREBASE_AUTH_DOMAIN=your_domain
VITE_FIREBASE_PROJECT_ID=your_project_id
```

```
VITE_FIREBASE_STORAGE_BUCKET=your_bucket
VITE_FIREBASE_MESSAGING_SENDER_ID=your_sender_id
VITE_FIREBASE_APP_ID=your_app_id
```

4. Start the development server:

```
npm run dev
```

Backend Setup (Flask)

1. Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate  # On Windows: .\venv\Scripts\activate
```

2. Install dependencies:

```
pip install -r requirements.txt
```

3. Add your Firebase credentials file:
- Download your Firebase service account key
- Save it as firebase_credentials.json in the root directory
4. Start the Flask server:

```
python app.py
```

Google Cloud Platform (GCP) Deployment

Frontend Deployment (Cloud Run)

1. Install Google Cloud CLI:
- Download from https://cloud.google.com/sdk/docs/install
2. Initialize GCP:

```
gcloud init
gcloud auth configure-docker
```

3. Build and push Docker image:

```
docker build -t gcr.io/[PROJECT_ID]/dinewise-frontend .
docker push gcr.io/[PROJECT_ID]/dinewise-frontend
```

4. Deploy to Cloud Run:

```
gcloud run deploy dinewise-frontend \
 --image gcr.io/[PROJECT_ID]/dinewise-frontend \
 --platform managed \
 --region [REGION] \
 --allow-unauthenticated
```

Backend Deployment (Cloud Run)

1. Build backend Docker image:

```
docker build -t gcr.io/[PROJECT_ID]/dinewise-backend -f Dockerfile.backend .
docker push gcr.io/[PROJECT_ID]/dinewise-backend
```

2. Deploy backend:

```
gcloud run deploy dinewise-backend \
  --image gcr.io/[PROJECT_ID]/dinewise-backend \
  --platform managed \
  --region [REGION] \
  --allow-unauthenticated
```

DigitalOcean Deployment

Prerequisites

- DigitalOcean account
- doctl CLI installed

Frontend Deployment (App Platform)

1. Install doctl and authenticate:

```
doctl auth init
```

2. Create app specification (app.yaml):

```
name: dinewise-frontend
region: nyc
services:
- name: web
  github:
    repo: your-repo
    branch: main
  source_dir: /
  envs:
  - key: NODE_ENV
    value: production
```

3. Deploy using App Platform:

```
doctl apps create --spec app.yaml
```

Backend Deployment (Droplet)

1. Create a Droplet:

```
doctl compute droplet create dinewise-backend \
  --image ubuntu-20-04-x64 \
  --size s-1vcpu-1gb \
  --region nyc1
```

2. SSH into the Droplet:

```
doctl compute ssh dinewise-backend
```

3. Install dependencies and setup:

```
sudo apt update && sudo apt upgrade -y
sudo apt install python3-pip nginx -y
git clone [your-repo-url]
cd dinewise
pip3 install -r requirements.txt
```

4. Setup Nginx and Gunicorn:

```
sudo nano /etc/nginx/sites-available/dinewise
```

Add configuration:

```
server {
  listen 80;
  server_name your_domain.com;

  location / {
    proxy_pass http://localhost:8000;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
  }
}
```

5. Enable and start services:

```
sudo ln -s /etc/nginx/sites-available/dinewise /etc/nginx/sites-enabled
sudo systemctl restart nginx
gunicorn -w 4 app:app
```

Important Security Notes

1. Always use HTTPS in production
2. Secure your environment variables
3. Keep Firebase credentials private
4. Regularly update dependencies
5. Set up monitoring and logging
6. Configure proper CORS settings

Troubleshooting

Common Issues

1. **Firebase Connection Issues**
   - Verify credentials file location
   - Check Firebase console permissions
2. **Deployment Failures**
   - Verify environment variables

   - Check build logs
   - Ensure sufficient resources
3. **Performance Issues**
   - Monitor resource usage
   - Check database indexes
   - Optimize API calls

For more detailed troubleshooting, consult the error logs in your respective platform's monitoring dashboard.