🚀 Deployment Manual

Flask + MySQL Application Using GitHub Codespaces

This document provides a complete step-by-step guide to deploy and run the Matrimony Flask backend inside GitHub Codespaces.

📌 Prerequisites

- GitHub account
- Project pushed to a GitHub repository
- Backup file (`backup.sql`) if restoring existing data(Already present in github)
- DOWNLOAD THE XGBOOST model(69.35%): https://drive.google.com/file/d/1qVf-fXvmZh6AqZUZQ1aNVcChINGVWWfP/view?usp=sharing
- Download the Encoder for columns: https://drive.google.com/file/d/1XuWcoFf5QIvUzEohGaWGZCR5NIizRzBB/view?usp=drive_link
- Create a resources folder under /FinalYearProject/Matrimony_Matchmaker/App/backend⇒ put the two files inside.
- Note: These last three steps are very crucial.

🟢 STEP 1 — Push Project to GitHub

Already done Nothing to do.

🟢 STEP 2 — Create GitHub Codespace

1. Open your repository on GitHub.
2. Click **Code**.
3. Select **Codespaces** tab.
4. Click **Create Codespace on main**.
5. Wait for the web-based VS Code environment to load.

🟢 STEP 3 — Setup Python Virtual Environment

Open terminal inside Codespace:

```
cd App/backend
```

you have `requirements.txt` :

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

Otherwise install manually:

```
pip install flask flask_sqlalchemy pymysql sqlalchemy
```

🟢 STEP 4 — Install MySQL Server

Update package manager:

```
sudo apt update
```

Install MySQL:

```
sudo apt install mysql-server -y
```

Start MySQL service:

```
sudo service mysql start
```

Check MySQL status:

```
sudo service mysql status
```

🟢 STEP 5 — Create Database

Enter MySQL shell:

```
sudo mysql
```

Inside MySQL:

```
CREATE DATABASE matrimony_db;
EXIT;
```

🟢 STEP 6 — Create Application Database User

⚠ Do NOT use root for application connections.

Enter MySQL again:

```
sudo mysql
```

Run:

```
CREATE USER 'matriuser'@'localhost' IDENTIFIED BY '1234';
GRANT ALL PRIVILEGES ON matrimony_db.* TO 'matriuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

🟢 STEP 7 — Restore Database Backup (Optional)

If you have `backup.sql` inside the Codespace:

```
sudo mysql -u matriuser -p matrimony_db < backup.sql
```

Enter password when prompted:

```
1234
```

🟢 STEP 8 — Configure Flask Database URI

Open `app.py` and set:

```
app.config['SQLALCHEMY_DATABASE_URI'] = \
    'mysql+pymysql://matriuser:1234@localhost/matrimony_db'
```

🟢 STEP 9 — Configure Flask Host & Port

Ensure Flask runs on:

```
app.run(debug=True, host="0.0.0.0", port=5000)
```

This is required for Codespaces port forwarding.

🟢 STEP 10 — Run Flask Application

Activate virtual environment (if not already active):

```
source venv/bin/activate
```

Start application:

```
python app.py
```

🟢 STEP 11 — Access Application

Codespaces will display:

> Port 5000 forwarded

Click:

Open in Browser

Your backend is now running successfully in the cloud.

🔧 Troubleshooting

If MySQL is not running:

```
sudo service mysql start
```

If Access Denied Error Appears:

Ensure you are using:

```
matriuser
```

Not:

```
root
```

Check MySQL Users:

```
sudo mysql
SELECT user, host FROM mysql.user;
```

🏗️ Final Architecture Overview

```
GitHub Codespace Environment
|
├── Flask Backend (Python 3.x)
├── Virtual Environment (venv)
├── MySQL Server (local container)
├── matrimony_db database
└── Application running on Port 5000
```

✅ Advantages of Using GitHub Codespaces

- Clean development environment
- No local configuration conflicts
- Cloud-based execution
- Easy collaboration
- Deployment-ready architecture
- Simplified migration to Docker / AWS / Render

📌 End of Deployment Manual
