Here is the **Deployment Plan** for your freelancing website project.

**📄 8. Deployment Plan**

**Project:** *FreelancerConnect*  
**Prepared By:** [Your Name]  
**Purpose:** To define the process, tools, and environments required to deploy the application safely and reliably.

**🧰 1. Deployment Strategy**

| **Type** | **Description** |
| --- | --- |
| **Rolling Deployment** | Gradual release of the new version to servers, minimizing downtime. |
| **Blue-Green Deployment (Optional)** | Two identical production environments — one live (green), one idle (blue). Deploy to blue, test, then switch. |

**🌐 2. Environments Setup**

| **Environment** | **Purpose** | **Hosting Platform** |
| --- | --- | --- |
| **Development** | Local testing (React + Spring Boot) | Localhost / Docker |
| **Staging** | Internal QA & UAT testing | AWS EC2 / Railway / Vercel |
| **Production** | Live user access | AWS / DigitalOcean / Vercel + PostgreSQL DB |

**⚙️ 3. Tech Stack Recap**

| **Layer** | **Stack** |
| --- | --- |
| Frontend | React + TailwindCSS |
| Backend | Spring Boot (REST APIs) |
| Database | PostgreSQL / MySQL |
| Authentication | JWT-based Auth |
| Storage | AWS S3 (for file uploads) |
| Payment | Razorpay / Stripe |
| CI/CD | GitHub Actions / Jenkins |
| Hosting | Vercel (React), EC2 (Java) |

**🔄 4. CI/CD Pipeline**

**📁 GitHub Actions Flow:**

on:

push:

branches:

- main

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- name: Set up JDK

uses: actions/setup-java@v3

with:

java-version: '21'

- name: Build with Maven

run: mvn clean install

deploy:

needs: build

runs-on: ubuntu-latest

steps:

- name: Deploy to EC2

run: |

scp -i key.pem target/app.jar ec2-user@<ip-address>:/home/ec2-user/app.jar

ssh -i key.pem ec2-user@<ip-address> "java -jar app.jar"

**For Frontend (React):**

Use **Vercel** or **Netlify**, which deploy on every commit.

**🚀 5. Deployment Steps**

**✅ For Backend (Spring Boot):**

1. Build JAR file: mvn clean package
2. SSH into EC2: ssh -i key.pem ec2-user@ip
3. Copy JAR: scp -i key.pem target/app.jar ec2-user@ip:/home/ec2-user/
4. Run: java -jar app.jar
5. Configure as systemd service (optional for auto-restart)

**✅ For Frontend (React):**

1. Build: npm run build
2. Deploy:
   * Vercel: Connect repo, auto-deploy on push.
   * Netlify: Drag & drop build/ folder or connect GitHub.

**🧪 6. Post-Deployment Checklist**

| **Task** | **Status** |
| --- | --- |
| Smoke test APIs | ✅ |
| Check frontend routes | ✅ |
| Validate auth & session flow | ✅ |
| Test payments in live mode | ✅ |
| Check console logs for errors | ✅ |
| Backup database | ✅ |

**🔒 7. Rollback Plan**

If something goes wrong:

* Keep a backup of the previous JAR file on server.
* Switch back to old version using:
* java -jar old-app.jar
* For frontend (Vercel/Netlify), revert last deployment via dashboard.

Would you like to continue with the final section:  
➡️ **Maintenance & Support Plan**?

If yes, reply **continue**.