Experiment No. 9

Aim: CI/CD Deployment with GitHub Actions + Render/Vercel

Steps: CI/CD with Render

Goal: Automatically deploy to Render when code is pushed to GitHub (main branch)

Step-by-Step Instructions:

1. Deploy your app on Render manually (first time)
   - Go to Render
   - Create a new Web Service or Static Site
   - Connect your GitHub repo OR set up manually
   - Wait until Render finishes the first deploy.

2. Create a Deploy Hook on Render
   1. Go to your service's Settings
   2. Scroll down to Deploy Hooks
   3. Click "Add Deploy Hook"
   4. Copy the Deploy Hook URL (you'll need this in the next step)

3. Add the Deploy Hook URL as a GitHub Secret
   1. Go to your GitHub repository
   2. Navigate to: Settings > Secrets and Variables > Actions
   3. Click "New repository secret"
   4. Name: RENDER_DEPLOY_HOOK_URL
      Value: (paste the hook URL you copied)

4. Add GitHub Actions Workflow
   1. In your project root, create the directory: .github/workflows
   2. Inside it, create a file named: deploy.yml
   3. Paste the following **code:**

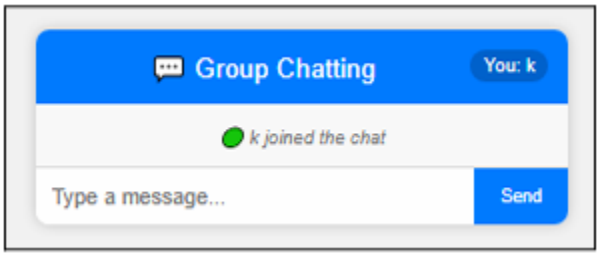
```yaml
name: Deploy to Render

on:
  push:
    branches:
      - master

jobs:
  deploy:
    runs-on: ubuntu-latest
```

```
steps:
  - name: Trigger Render Deploy Hook
    run: |
          curl -X POST ${{ secrets.RENDER_DEPLOY_HOOK_URL }}
```

Now every time you push to the main branch, GitHub Actions will trigger a new deployment on Render

Code:

Output: Before push on git



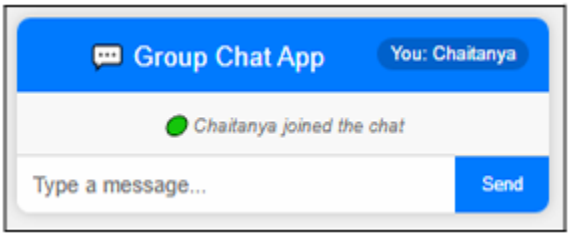
Push updated code to git

```
D:\GroupChat>git add .

D:\GroupChat>git commit -m "new add"
[master d06158d] new add
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\GroupChat>git add .

D:\GroupChat>git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 397 bytes | 198.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/mr-chaitanyad/GroupChatApp.git
   8869484..d06158d  master -> master
```

On web services logs on render

```
Oct 6 11:20:38 PM  INFO          ==> Deploying...
Oct 6 11:20:43 PM  INFO    v8gvh  ● New user connected
Oct 6 11:20:57 PM  INFO    x6lcn  ==> Running 'npm start'
Oct 6 11:20:58 PM  INFO    x6lcn
Oct 6 11:20:58 PM  INFO    x6lcn  > backend@1.0.0 start
Oct 6 11:20:58 PM  INFO    x6lcn  > node server.js
Oct 6 11:20:58 PM  INFO    x6lcn
Oct 6 11:21:00 PM  INFO    x6lcn  🚀 Server running on http://localhost:5000
Oct 6 11:21:09 PM  INFO          ==> Your service is live 🎉
Oct 6 11:21:09 PM  INFO          ==>
Oct 6 11:21:09 PM  INFO          ==> ///////////////////////////////////////
Oct 6 11:21:09 PM  INFO          ==>
Oct 6 11:21:09 PM  INFO          ==> Available at your primary URL https://g
Oct 6 11:21:09 PM  INFO          ==>
Oct 6 11:21:09 PM  INFO          ==> ///////////////////////////////////////
Oct 6 11:21:22 PM  INFO    x6lcn  ● New user connected
```

After push on git



Conclusion:

In this setup, we integrated GitHub Actions with Render using a Deploy Hook to enable automatic deployment on every push to the main/master branch. This creates a simple and secure CI/CD pipeline, ensuring your app is always up to date without manual deployment.