Creating Web Apps

MGMT 675: Generative AI for Finance

Kerry Back

What is Streamlit?

Streamlit Overview

- Python library for web apps
- No HTML, CSS, or JavaScript needed
- Designed for data apps and dashboards
- Interactive widgets built-in
- Hot reload during development

Perfect For

- Data visualization dashboards
- Machine learning demos
- Financial analysis tools
- Interactive reports
- Chatbot interfaces

From Idea to Public App

The complete workflow to create and deploy a Streamlit app—all handled by Claude Code with simple natural language requests.

1. Create the Streamlit app
2. Initialize a Git repository
3. Create a GitHub repository
4. Commit and push the code
5. Deploy to Koyeb with auto-deploy
6. Get your public URL

Ask Claude to do each step—no commands to memorize

Step 1: Create the Streamlit App

What You Need

- Main app file (`app.py`)
- Dependencies file (`requirements.txt`)
- Any data files or assets

Ask Claude

"Create a Streamlit app that [describe your app]"

- Claude writes the Python code
- Creates requirements.txt automatically
- Can test locally first
- Iterates based on your feedback

Step 2: Initialize Git Repository

What Happens

- Creates `.git` folder in your project
- Enables version control
- Tracks all file changes
- Creates `.gitignore` for Python

Ask Claude

"Initialize this folder as a git repository"

- Claude runs `git init`
- Creates appropriate .gitignore
- Ready for commits

Step 3: Create GitHub Repository

What Happens

- Creates new repo on GitHub.com
- Uses GitHub CLI (`gh`)
- Links local repo to remote
- Can be public or private

Ask Claude

"Create a GitHub repository for this project called [name]"

- Claude uses `gh repo create`
- Sets up remote origin
- Ready to push code

Step 4: Commit and Push

What Happens

- Stages all project files
- Creates initial commit
- Pushes to GitHub
- Code now on GitHub.com

Ask Claude

"Commit all files and push to GitHub"

- Claude runs git add, commit, push
- Writes meaningful commit message
- Verifies push succeeded

Step 5: Deploy to Koyeb

What is Koyeb?

- Cloud platform for deploying apps
- Free tier available
- Auto-deploy from GitHub
- Handles SSL, scaling, etc.

Ask Claude

"Create a Koyeb service for this app linked to the GitHub repo with auto-deploy"

- Claude uses Koyeb CLI
- Links to GitHub repo
- Enables auto-deploy on push

Step 6: Get Your Public URL

What Happens

- Koyeb provides public URL
- Format: `app-name.koyeb.app`
- SSL included automatically
- App accessible worldwide

Ask Claude

"What is the public URL for my Koyeb service?"

- Claude queries Koyeb CLI
- Returns your public URL
- You can share with anyone

Auto-Deploy: The Magic

Once set up, every `git push` automatically updates your live app.

Update Workflow

1. Make changes locally
2. Ask Claude to commit and push
3. Koyeb detects the push
4. Rebuilds and redeploys
5. Live app updated automatically

- No manual deployment steps
- Changes live in minutes
- Rollback if needed
- View deployment logs

Exercise: Mean-Variance App

Build and deploy a Streamlit app for mean-variance portfolio optimization.

User Inputs

- Risk-free rate
- Number of risky assets
- Expected returns (means)
- Standard deviations
- Correlation matrix

App Outputs

- Tangency portfolio weights
- Downloadable image of:
  - Frontier of risky assets
  - Capital allocation line

Summary: One Conversation, Live App

Step	Ask Claude
Create app	"Create a Streamlit app that..."
Init git	"Initialize as a git repository"
Create GitHub repo	"Create a GitHub repo called..."
Commit & push	"Commit and push to GitHub"
Deploy	"Deploy to Koyeb with auto-deploy"
Get URL	"What's my public URL?"

No commands to memorize. Just describe what you want.