1. Project Setup

1. Create a new folder on your computer named "ik-dashboard"
2. Save your current dashboard code as app.py in this folder
3. Create a new file named requirements.txt in the same folder
4. Add these lines to requirements.txt:

text

streamlit

pandas

plotly

openpyxl

2. Data Handling

1. Create a new folder named "data" in your project folder
2. Copy your Excel file into this folder
3. Update the file path in your code to use a relative path:

python

df = pd.read\_excel("data/IK\_Konj+Destatis\_HWWI.xlsx")

3. GitHub Setup

1. Go to github.com and create an account
2. Download and install Git from git-scm.com
3. Open Command Prompt/Terminal
4. Navigate to your project folder:

bash

cd path/to/ik-dashboard

1. Initialize Git repository:

bash

git init

1. Create .gitignore file and add:

text

\*.xlsx

\_\_pycache\_\_/

.env

1. Add your files:

bash

git add app.py requirements.txt .gitignore

1. Commit your files:

bash

git commit -m "Initial commit"

4. GitHub Repository Creation

1. Go to github.com
2. Click "+" in top right corner
3. Select "New repository"
4. Name it "ik-dashboard"
5. Keep it private
6. Click "Create repository"
7. Follow the instructions to push your code:

bash

git remote add origin [your-repository-URL]

git branch -M main

git push -u origin main

5. Streamlit Cloud Deployment

1. Go to share.streamlit.io
2. Sign in with your GitHub account
3. Click "New app"
4. Select your repository "ik-dashboard"
5. Select main branch
6. Select app.py as the main file
7. Click "Deploy"

6. Data Security

1. Consider moving sensitive data to a secure database
2. Update your code to use environment variables for sensitive information
3. Test the app thoroughly before sharing

7. Sharing

1. Get the public URL from Streamlit Cloud
2. Test the dashboard thoroughly
3. Share the URL with your colleagues

8. Maintenance

1. Keep your local repository updated
2. Make changes locally and test
3. Push updates to GitHub:

bash

git add .

git commit -m "Update description"

git push

1. Streamlit Cloud will automatically update your app

The dashboard will remain accessible 24/7 and update automatically when you push changes to GitHub.

**GitHub Hosting**

Unter diesem Link müssen Code und Dateien etc. aktualisiert werden, parallel dazu Aktualisierung auf dem Laufwerk der IK [IK-Coding2025/ik-dashboard](https://github.com/IK-Coding2025/ik-dashboard)

**Start Git Hub Bash**

Rechtsklick mit Maus im Ordner:  
Ein Bild, das Text, Screenshot, Display, Computer enthält.

Automatisch generierte Beschreibung

Dann im aufpoppenten Fenster von GitHub programmieren:

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main|MERGING)

$ "C:\Users\l.mueller\Documents\FileCloud\Team Folders\IK\_Server\Wirtschaft\statistische Daten\ik-dashboard"

bash: C:\Users\l.mueller\Documents\FileCloud\Team Folders\IK\_Server\Wirtschaft\statistische Daten\ik-dashboard: Is a directory

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main|MERGING)

$ git add .

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main|MERGING)

$ git commit -m "Update app with plotly dpendency"

[main ebcf21a] Update app with plotly dpendency

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main)

$ git push

Enumerating objects: 8, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 12 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (6/6), 108.84 KiB | 27.21 MiB/s, done.

Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (2/2), completed with 1 local object.

To https://github.com/IK-Coding2025/ik-dashboard.git

1aa7f91..ebcf21a main -> main

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main)

$ git add assets/

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main)

$ git commit -m "Add assets folder with IK logo"

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main)

$ git push origin main

Everything up-to-date

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main)

$ git push origin main

To https://github.com/IK-Coding2025/ik-dashboard.git

! [rejected] main -> main (fetch first)

error: failed to push some refs to 'https://github.com/IK-Coding2025/ik-dashboard.git'

hint: Updates were rejected because the remote contains work that you do not

hint: have locally. This is usually caused by another repository pushing to

hint: the same ref. If you want to integrate the remote changes, use

hint: 'git pull' before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische Daten/ik-dashboard (main)

$ git pull

remote: Enumerating objects: 21, done.

remote: Counting objects: 100% (21/21), done.

remote: Compressing objects: 100% (19/19), done.

remote: Total 19 (delta 12), reused 0 (delta 0), pack-reused 0 (from 0)

Unpacking objects: 100% (19/19), 6.43 KiB | 177.00 KiB/s, done.

From https://github.com/IK-Coding2025/ik-dashboard

ebcf21a..c0f43dc main -> origin/main

Updating ebcf21a..c0f43dc

Fast-forward

app.py | 14 +++++++-------

1 file changed, 7 insertions(+), 7 deletions(-)

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git push origin main

Everything up-to-date

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git add data/

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git commit -m "Update data folder with Excel fildes"

[main c30f4a8] Update data folder with Excel fildes

1 file changed, 0 insertions(+), 0 deletions(-)

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git commit -m "Update data folder with Excel files"

On branch main

Your branch is ahead of 'origin/main' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git push origin main

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 12 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 756 bytes | 756.00 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/IK-Coding2025/ik-dashboard.git

c0f43dc..c30f4a8 main -> main

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git add data/

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git comit -m "Update data folder with Excel files"

git: 'comit' is not a git command. See 'git --help'.

The most similar command is

commit

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git commit -m "Update data folder with Excel files"

[main 5a05659] Update data folder with Excel files

1 file changed, 0 insertions(+), 0 deletions(-)

l.mueller@Mueller-NB MINGW64 ~/Documents/FileCloud/Team Folders/IK\_Server/Wirtschaft/statistische

$ git push origin main

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 12 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 13.16 KiB | 13.16 MiB/s, done.

Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/IK-Coding2025/ik-dashboard.git

c30f4a8..5a05659 main -> main