Hackathon Preparation Workshop

Brought to you by CompSoc and ShefESH

Session Overview

- Timetable
- What is a hackathon? How to do a perfect pitch
- Ideal workflow
- Tools:
  - Git
  - Python + Supabase
  - Flask
  - React
    - Tailwind
- Devpost
- Do and Don’ts
- What to bring

Timetable

Saturday

09:00	Doors Open
10:00	Team Building
10:30 - 12:00	Opening Ceremony
12:00	Hacking Begins!
13:00	Lunch
14:00 - 14:45	Reply: Agentic AI Workshop
15:00 - 15:30	Introduction to Gafana

16:00	ShefESH: Capture The Flag
17:00 - 18:30	Carnival Games
19:00	Dinner
20:00 - 21:00	TechVision: Leetcode challenge
21:00 - 22:00	GameDev Soc: Gamification Mini Competition
23:00	Group Minigame: Werewolf

Sunday

09:00	Morning Snacks
10:00	Morning Walk
12:00	Hacking Closes + Lunch
13:00 - 17:00	Judging + Closing

What is a hackathon?

What is it?

- An event where people come together to solve challenges through
  - intense software development,
  - rapid prototyping, and
  - working collaboratively

- **Goal:** To prototype a product surrounding a theme, and persuade the panel of judges with your pitch

What is it?

- HackSheff: 24 hour Hackathon
- It will be quite competitive, some stay up all night working on this
- There will be a list of themes,
  - Pick one and aim to win in that

Pitch is as important as your product

How to Pitch?

- Before pitch, aim to get a MVP (Minimal Viable Product) instead of a perfect product
- You will have roughly **3** minutes for your presentation

How to Pitch?

- Aim for:
  - **Short, captivating hook** -> have a compelling narrative linked to the theme, keep it grounded, introduce problem
  - **Present your solution** -> have a demo, show its uniqueness and strengths
  - **Explain its impact** and talk about decisions you made in the prototyping phase
  - **Know what judges are looking** for, tailor to it
  - Aim for a **strong finish**, the start and end is what they will stick with them

Important: Make it unique and interesting, there will be min 50+ team presenting, try to stand out and be memorable

Saturday

09:00	Doors Open
10:00	Team Building
10:30 - 12:00	Opening Ceremony
12:00	Hacking Begins!
13:00	Lunch
14:00 - 14:45	Reply: Agentic AI Workshop
15:00 - 15:30	Introduction to Gafana

Brainstorm

Project setup

16:00	ShefESH: Capture The Flag
17:00 - 18:30	Carnival Games
19:00	Dinner
20:00 - 21:00	TechVision: Leetcode challenge
21:00 - 22:00	GameDev Soc: Gamification Mini Competition
23:00	Group Minigame: Werewolf

Building

Prototyping

Ideal Workflow

Final Touches

Focus on presentation

09:00	Morning Snacks
10:00	Morning Walk
12:00	Hacking Closes + Lunch
13:00 - 17:00	Judging + Closing

Ideal Workflow

- Brainstorming & Deciding Roles - **roughly 1-1:30 hrs**
  - Define the problem, discuss ideas, scope & align team
- Project Setup - **30 mins**
  - Setup repo, assign role, define MVP features (use issues for project management)
- Building - **60%**
  - Core application logic, basic UI
- Prototyping - **40 %**
  - Integrate components, basic testing
  - Useable demo by the end of day 1
- Refining for theme - **2 hours**
- Making presentation & Practice Pitch - **1- 1:30 hrs**

Tools

Example Tech Stack

Example

- Front-end: React (JavaScript)
- Back-end: Flask (Python)
- Data Storage + Auth: Supabase (PostgreSQL)
- Version-Control: GitHub

Git and GitHub

What is Git?

- Git is a popular version control system used for tracking code changes, who made them and code collaboration
- Things you can do with git:
  - Create a repository by initialising git on a folder
  - Commit modifications to files by pushing updates
  - Pull the latest version of files to a local copy
  - Revert to previous commits
  - Branch and merge to allow for work on different sections/versions at the same time

Installing Git

There are a few different ways of accomplishing this:

- By installing GitHub Desktop
- By installing from the Internet (for Windows/Mac)
- By installing through VS Code GitHub Pull Requests and Issues extension
- Mac specific: Using Homebrew
- Debian/Ubuntu specific: running ‘sudo apt-get install git-all’
- Other Linux: you should know how to install git

Configuring Git

This is an important step to be able to commit file updates as it lets Git know who you are, and is done by running the following in Git Bash (for windows) or terminal (for Mac/Linux):

- git config --global user.name “your username”
- git config --global user.email “your email”

The email should be the same as the email you use/will use for GitHub

Creating a repository

To begin with, create an empty folder and then navigate to it within Bash/Terminal using the ‘cd’ command from COM1001 e.g. cd Documents/folder name

Once you are within the folder, you need to run the command ‘git init’ to initialise Git on that folder

If this is successful, you should get a message returned to you saying ‘Initialized empty Git repository in (place your folder is)’

Adding a file

Within your folder, create and save a new text document using a text editor such as Microsoft Word or Notepad with some information e.g. “Hello World”

Return back to Bash/Terminal and type ‘git status’

```
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Hello.txt

nothing added to commit but untracked files present (use "git add" to track)
```

This sho e we just add

Staging a file

Now, we can use the command ‘git add (text file)’ to stage the file, which means that the file is ready to be committed.

If we had multiple files to stage, we can use the command ‘git add --all’ or ‘git add -A’

To check this has the ‘git status’ command again

```
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Hello.txt
```

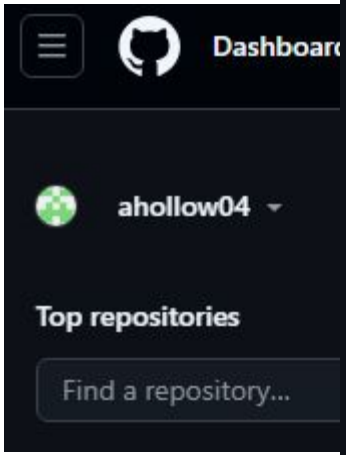
Committing a file

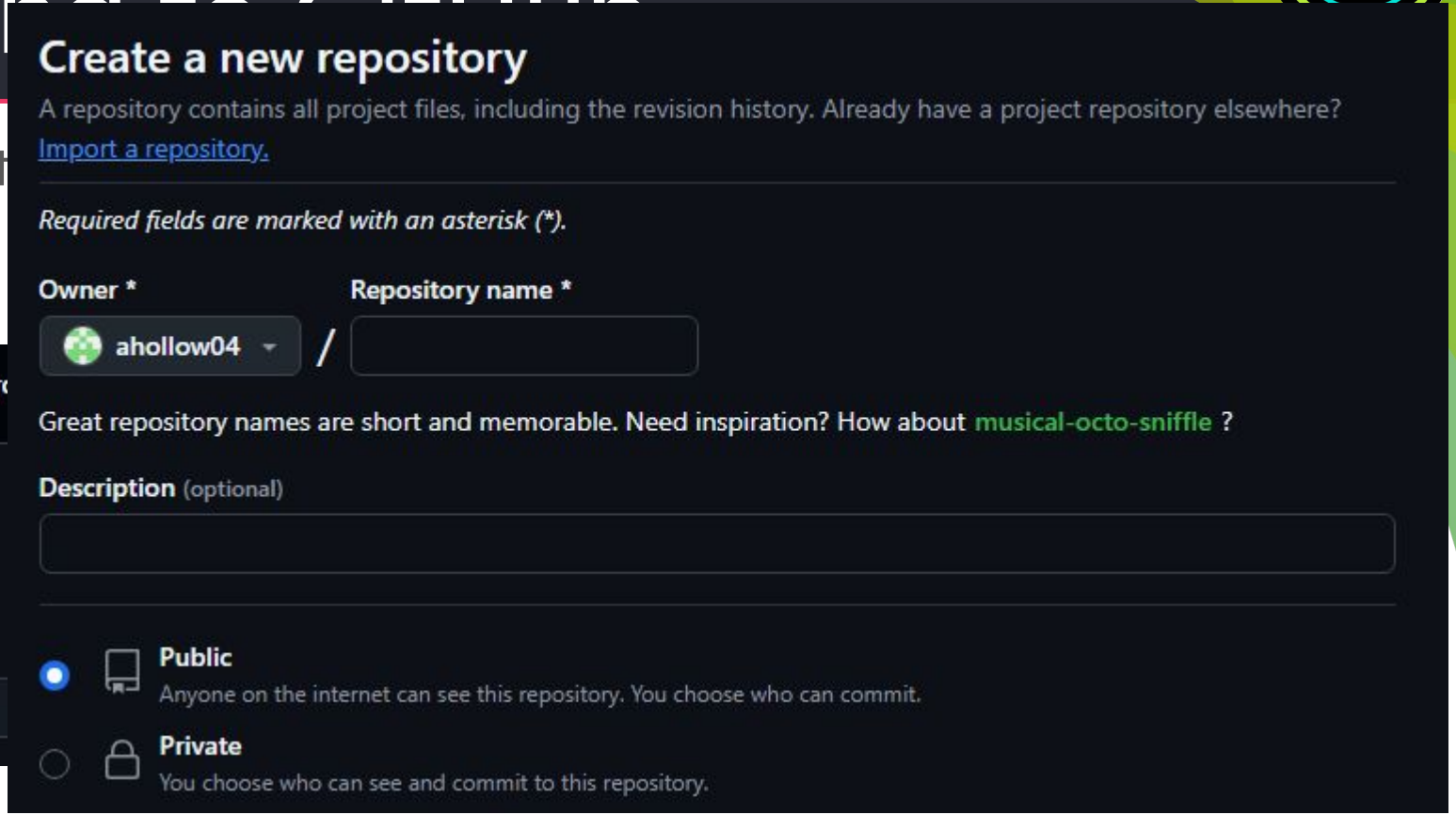
When we commit files, it is important to always include a clear message to help identify to yourself and others what has changed and when.

This is done using the command: ‘git commit -m “Useful message here”’

Pushing to GitHub

To be able t
repository:





Pushing to GitHub

You will need to copy the URL and use it in the following command: 'git remote add origin (paste URL)'

Pushing to GitHub

Now that you have set up a connection between your local Git repository and your online GitHub repository, you can now run the following command: ‘git push --set-upstream origin master’ (sometimes main instead of master)

Since this is the first time you are pushing to GitHub, you need to use ‘--set-upstream’ to identify the default branch you want to push to

If you refresh your GitHub page, you should see that your repository has updated

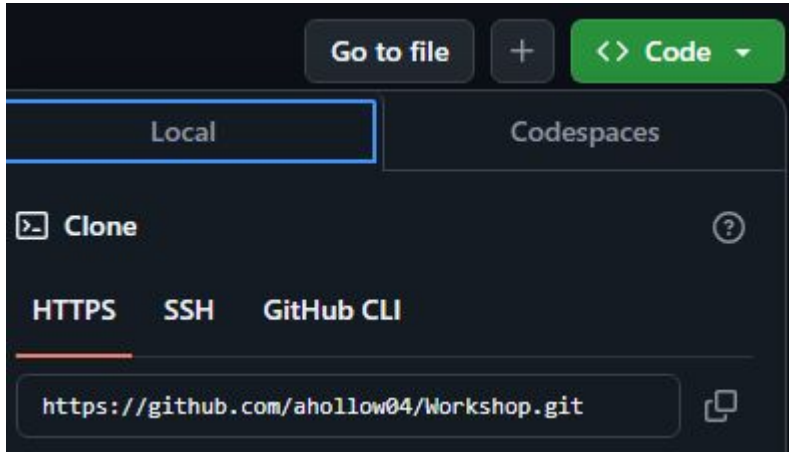
Pulling from GitHub

This is used to update your local version of a repository by using the command ‘git pull origin’

By using pull, we are using both fetch and merge commands behind the scenes, where fetch gets all of the change history and merge combines the current branch with a specified branch.

What if you want to work on an existing repository?

This can be done by cloning an existing repository so that you can work on it locally, using the command: ‘git clone (URL)’ where you can copy the URL from:



Updates can be committed using:

- git add (file name)
- git commit -m “”
- git push

Branches

These are extremely useful to work on new features of a project in a contained area of the repository, ensuring that any breakages to code only affect the project branch and not the project itself.

Another benefit of branches is that multiple developers can work on separate tasks at the same time without causing multiple project conflicts.

Pushing a branch to GitHub

In order to create a new branch, we use the following command: ‘git checkout -b (new branch name)’

Afterwards, make a couple of changes to the text file e.g. adding another word (don’t forget to save!)

Now, che us’ - it is importo our new bran

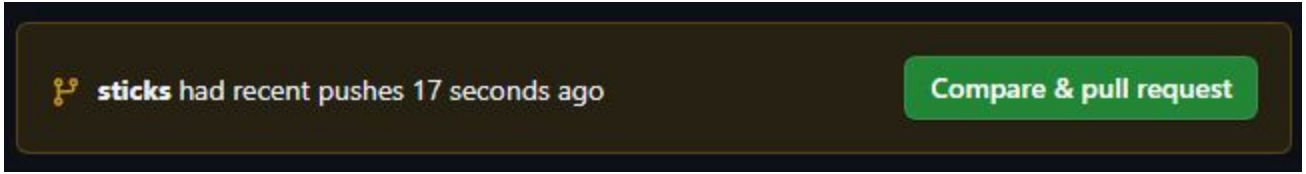
```
On branch sticks
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

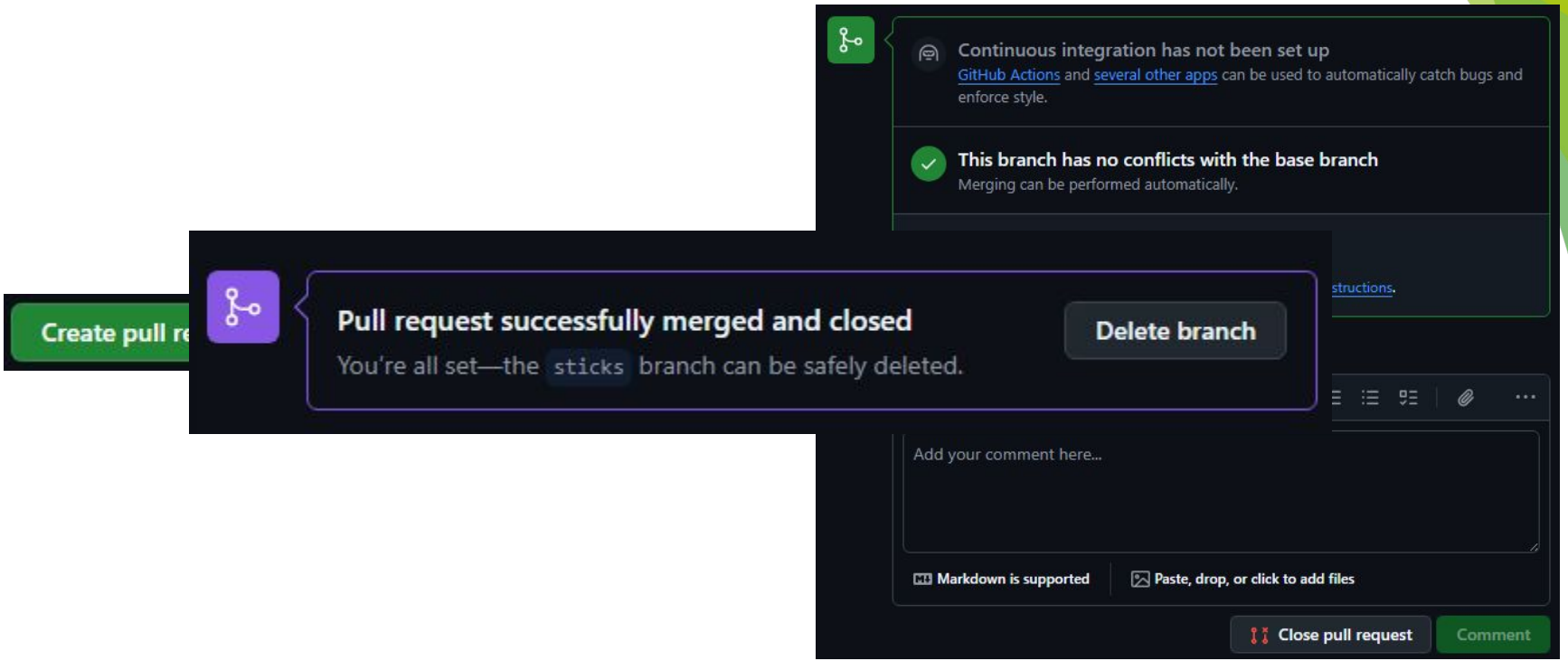
Pushing a branch to GitHub

Like before, we can now use ‘git add (file name)’ and ‘git commit -m “(useful message)”’

Now, to push the newly created branch, we use the command ‘git push origin (new branch name)’



Pushing a branch to GitHub



Pulling a branch from GitHub

When a branch has been added to a GitHub repository, it will show up when you run ‘git pull’ e.g.

```
From https://github.com/ahollow04/Workshop
   8259aa2..8d00d0b  master     -> origin/master
 * [new branch]      sticks     -> origin/sticks
```

We can find out what branches are available locally and remotely by using ‘git branch -a’

```
* master
  remotes/origin/master
  remotes/origin/sticks
```

And in order access remote branches locally we use ‘git checkout (branch name)’

```
Switched to a new branch 'sticks'
branch 'sticks' set up to track 'or
```

```
  master
* sticks
  remotes/origin/master
  remotes/origin/sticks
```

Python

Python

- A language used by almost everyone
- Very beginner friendly
- Also very powerful

Projects

- AI / ML
- Websites
- Data Science
- Simple GUI apps

Python venv

PIP sometimes breaks

- System package issues
- Version issues

To fix this, we use **v**irtual **env**ironments!

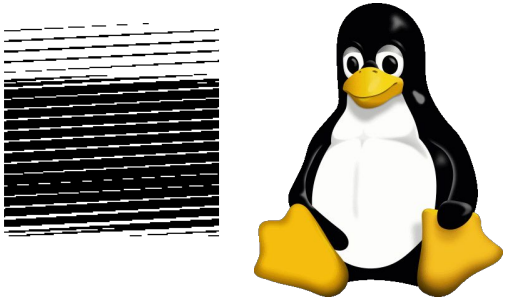
https://docs.python.org/3/library/venv.html

Keeps Python packages separate

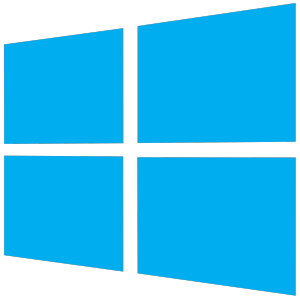
- Less likely to break something else

```
/venv-example $ python -m venv virtualenv
```

```
VENV-EXAMPLE
> virtualenv
```



```
source virtualenv/bin/activate
```



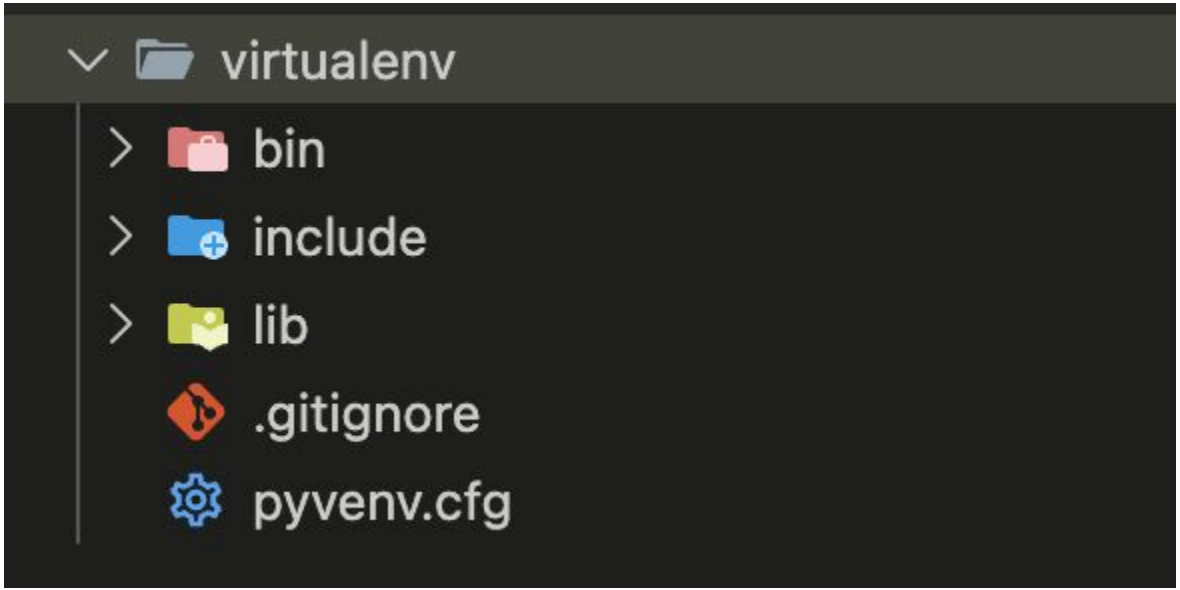
```
./virtualenv/Scripts/activate
```

Unactivated:

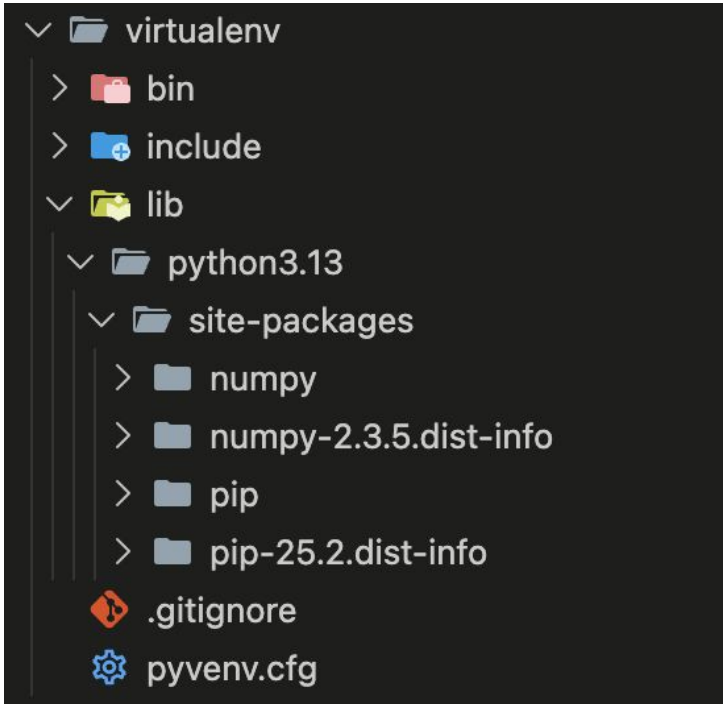
```
oliver@100.65.210.15 13:54 -> ~/Desktop/Dev/Python/venv-example $
```

Activated:

```
(virtualenv) oliver@100.65.210.15 13:54 -> ~/Desktop/Dev/Python/venv-example $
```


virtualenv
bin
include
lib
.gitignore
pyvenv.cfg

```
(virtualenv) oliver@100.65.210.15 13:54 -> ~/Desktop/Dev/Python/venv-example $ pip install numpy
Collecting numpy
  Using cached numpy-2.3.5-cp313-cp313-macosx_14_0_arm64.whl.metadata (62 kB)
Using cached numpy-2.3.5-cp313-cp313-macosx_14_0_arm64.whl (5.1 MB)
Installing collected packages: numpy
Successfully installed numpy-2.3.5
```


virtualenv
bin
include
lib
python3.13
site-packages
numpy
numpy-2.3.5.dist-info
pip
pip-25.2.dist-info
.gitignore
pyvenv.cfg

Requirements.txt

Will definitely be using Python libraries

More you use the harder it gets to keep track

- Where requirements.txt comes in

Text file containing all the libraries (and versions if needed)

requirements.txt

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

requirements.txt

```
numpy
matplotlib
bs4
Pillow
requests>2.0

pandas==2.3.3
```

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

```
(virtualenv) oliver@100.65.210.15 14:07 -> ~/Desktop/Dev/Python/venv-example $ pip install -r requirements.txt
Collecting numpy (from -r requirements.txt (line 1))
  Using cached numpy-2.3.5-cp313-cp313-macosx_14_0_arm64.whl.metadata (62 kB)
Collecting matplotlib (from -r requirements.txt (line 2))
  Using cached matplotlib-3.10.7-cp313-cp313-macosx_11_0_arm64.whl.metadata (11 kB)
Collecting bs4 (from -r requirements.txt (line 3))
  Using cached bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Collecting Pillow (from -r requirements.txt (line 4))
  Using cached pillow-12.0.0-cp313-cp313-macosx_11_0_arm64.whl.metadata (8.8 kB)
Collecting requests>2.0 (from -r requirements.txt (line 6))
  Using cached requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting pandas==2.3.3 (from -r requirements.txt (line 8))
  Using cached pandas-2.3.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (91 kB)
Collecting python-dateutil>=2.8.2 (from pandas==2.3.3->-r requirements.txt (line 8))
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas==2.3.3->-r requirements.txt (line 8))
  Using cached pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas==2.3.3->-r requirements.txt (line 8))
  Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting contourpy>=1.0.1 (from matplotlib->-r requirements.txt (line 2))
  Using cached contourpy-1.3.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib->-r requirements.txt (line 2))
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib->-r requirements.txt (line 2))
  Using cached fonttools-4.60.1-cp313-cp313-macosx_10_13_universal2.whl.metadata (112 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib->-r requirements.txt (line 2))
  Using cached kiwisolver-1.4.9-cp313-cp313-macosx_11_0_arm64.whl.metadata (6.3 kB)
Collecting packaging>=20.0 (from matplotlib->-r requirements.txt (line 2))
  Using cached packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting pyparsing>=3 (from matplotlib->-r requirements.txt (line 2))
  Using cached pyparsing-3.2.5-py3-none-any.whl.metadata (5.0 kB)
Collecting beautifulsoup4 (from bs4->-r requirements.txt (line 3))
  Using cached beautifulsoup4-4.14.2-py3-none-any.whl.metadata (3.8 kB)
Collecting charset-normalizer<4,>=2 (from requests>2.0->-r requirements.txt (line 6))
```

```
  Using cached pillow-12.0.0-cp313-cp313-macosx_11_0_arm64.whl.metada
Collecting requests>2.0 (from -r requirements.txt (line 6))
  Using cached requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting pandas==2.3.3 (from -r requirements.txt (line 8))
```

```
  Using cached requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting pandas==2.3.3 (from -r requirements.txt (line 8))
  Using cached pandas-2.3.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (91 kB)
Collecting python-dateutil>=2.8.2 (from pandas==2.3.3->-r requirements.txt (lin
```

Libraries

Python is full of useful libraries:

- MatPlotLib - graphs
- Numpy - maths
- Pandas - big data
- scikit-learn - machine learning
- Pytorch - machine learning
- Kivy - GUI

Pick and choose what works for your project. Use libraries often, it saves you time!

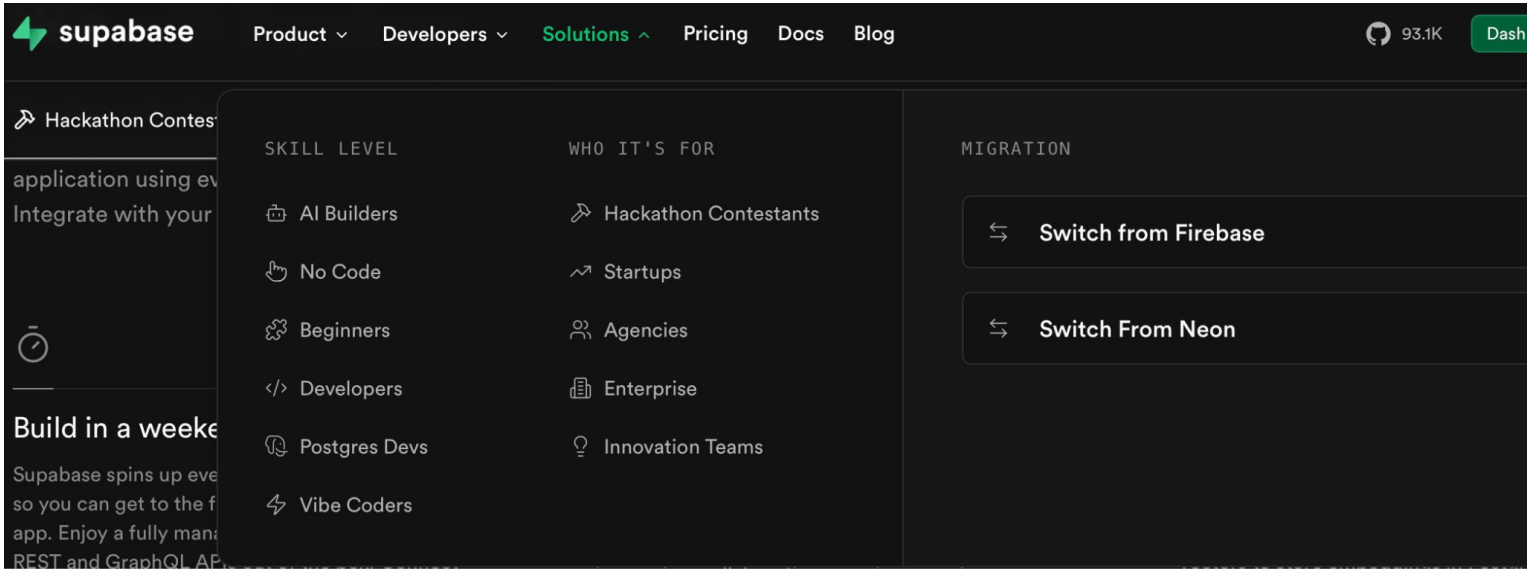
Supabase

What is it?

Backend as a Service

- Tools to handle common backend tasks without having to build them from scratch
- Helps save development time
- Don’t have to reinvent the wheel

For Hackathon Contestants!



Features

Full database
- Powered by PostGres
  - Industry database
  - Solid + Secure

Authentication
- Email + Password
- Magic link
  - (Click a link in an email)

Realtime
- Websockets
  - Database changes (i.e realtime stock updates)
  - Channels (i.e realtime chat)

Features (Cont.)

Storage

- Upload images/files/anything
  - Stored in ‘buckets’
  - Allows for image modification (i.e for preview images)

Extras

- Vectors (Machine learning embeddings, similarity searches)
- Cron jobs (Run certain functions at certain intervals / times)
- Queues (i.e Ensuring messages arrive in the order sent)

How to use

Host locally (via docker)

- Free + no limits
- Harder to use in a team

Free tier

- Some limits (very generous)
- Whole team can use (hosted remotely)
- Email sending included

Loads of languages

Javascript	Flutter	Python
C#	Swift	Kotlin

React

React

What is React?

- It is your front-end
- A JavaScript library for building user interfaces using declarative, component-based views.
- Open-source, widely used = lots of help online, easy to work with

- Apps built using react:

How to set it up

- Install node.js

- npm create vite@latest
  - Enter project-name
  - Framework: React
  - Js vs Ts
  - Rolldown-vite
  - Install with npm
  - Choose default options
- npm install react-router-dom

```
PS C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep> npm create vite@latest

> npx
> create-vite

◇ Project name:
  hackathon-prep-sesh

◇ Select a framework:
  React

◇ Select a variant:
  JavaScript

◇ Use rolldown-vite (Experimental)?:
  No

◇ Install with npm and start now?
  Yes

◇ Scaffolding project in C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep\hackathon-

◇ Installing dependencies with npm...

added 203 packages, and audited 204 packages in 9s

33 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

React Component Libraries

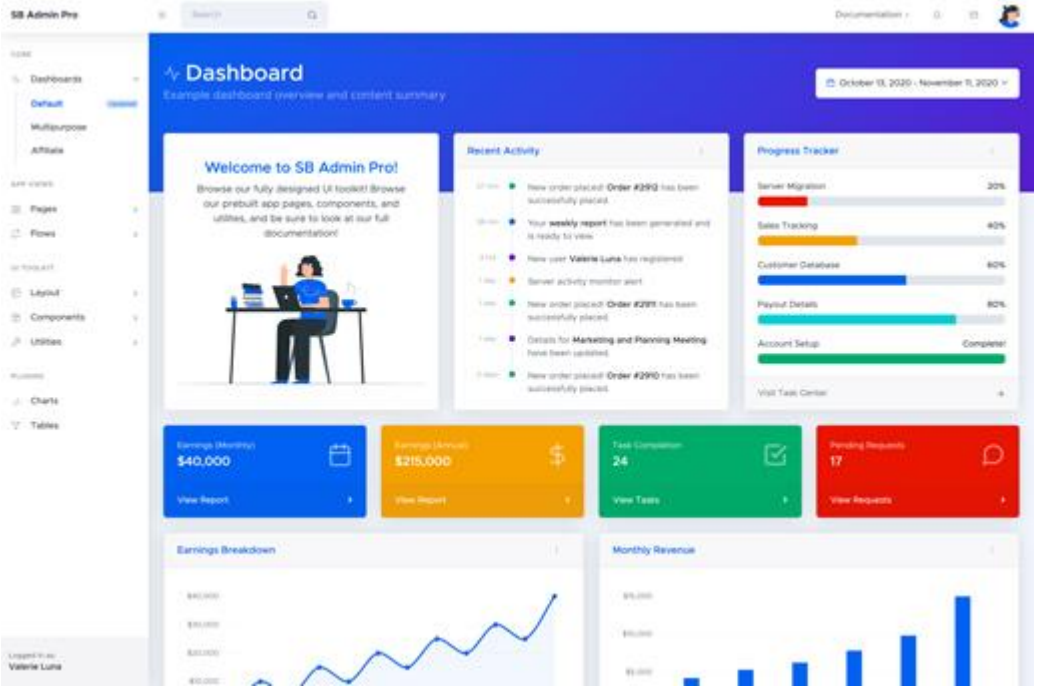
- Bootstrap
  https://react-bootstrap.netlify.app/docs/getting-started/introduction

- Material UI (MUI)
  https://mui.com/material-ui/all-components/

- Untitled UI
  https://www.untitledui.com/react/components

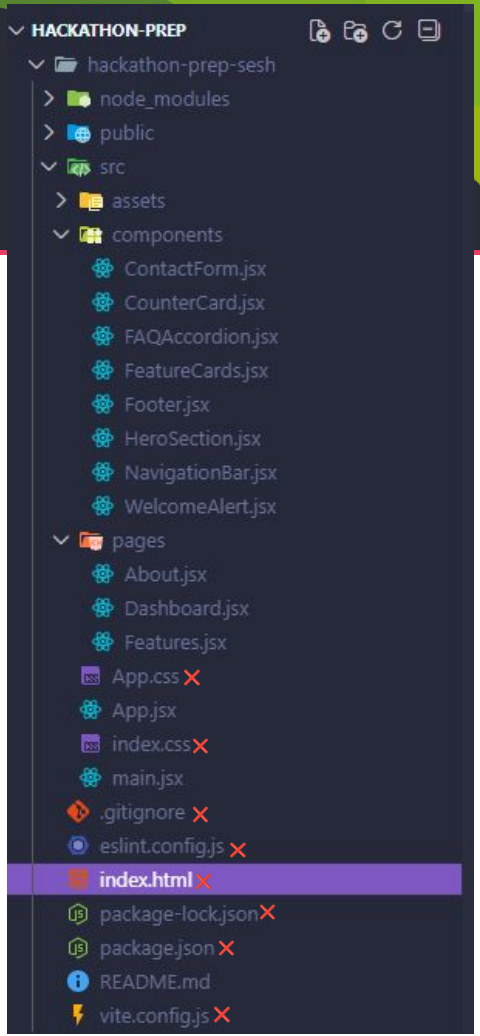
Bootstrap

- npm install react-bootstrap bootstrapz - website



React - File Structure

- index.html
  - Main.jsx ← main entry point of application
    - App.jsx ← holds different routes
      - <page_name>.jsx (eg. Dashboard.jsx)
        - components/
          - ContactForm.jsx
          - CounterCard.jsx
          - ...
          - WelcomeAlert.jsx

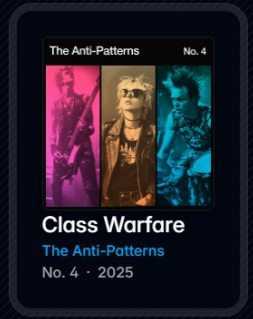


Tailwind

What is Tailwind?

- Easy way to style
- Utility-first CSS framework for building custom designs quickly without writing custom CSS

```html
<div class="flex flex-col items-center p-7 rounded-2xl">
  <div>
    <img class="size-48 shadow-xl rounded-md" alt="" src="/img/cover.png" />
  </div>
  <div class="flex">
    <span class="text-2xl font-medium">Class Warfare</span>
    <span class="font-medium text-sky-500">The Anti-Patterns</span>
    <span class="flex gap-2 font-medium text-gray-600 dark:text-gray-400">
      <span>No. 4</span>
      <span>·</span>
      <span>2025</span>
    </span>
  </div>
</div>
```



How to set it up

- npm install tailwindcss @tailwindcss/vite
- Add code to vite.config
- Add @import “tailwindcss/vite” to app.css

02 Install Tailwind CSS

Install `tailwindcss` and `@tailwindcss/vite` via npm.

Terminal

```
npm install tailwindcss @tailwindcss/vite
```

03 Configure the Vite plugin

Add the `@tailwindcss/vite` plugin to your Vite configuration.

vite.config.ts

```
import { defineConfig } from 'vite'
import tailwindcss from '@tailwindcss/vite'

export default defineConfig({
  plugins: [
    tailwindcss(),
  ],
})
```

04 Import Tailwind CSS

Add an `@import` to your CSS file that imports Tailwind CSS.

CSS

```
@import "tailwindcss";
```

05 Start your build process

Run your build process with `npm run dev` or whatever command is configured in your `package.json` file.

Terminal

```
npm run dev
```

How to use it

- React: className = “<css-code>”
- Download “Tailwind CSS IntelliSense” on VSCode
- Eg:
  - text-<color>-strength(100,200…900) ← Text color
  - bg-<color>-strength(100,200…900) ← background color
  - Text-4x1 ← Text Size
  - https://tailwindcss.com/docs/styling-with-utility-classes

Flask

Flask

What is Flask?

- A web framework for Python
- Simple and lightweight

- Allows you a lot of control over exactly how it will function
- Packaged with a webserver so can be ran on your device (Only for use in testing)

Flask

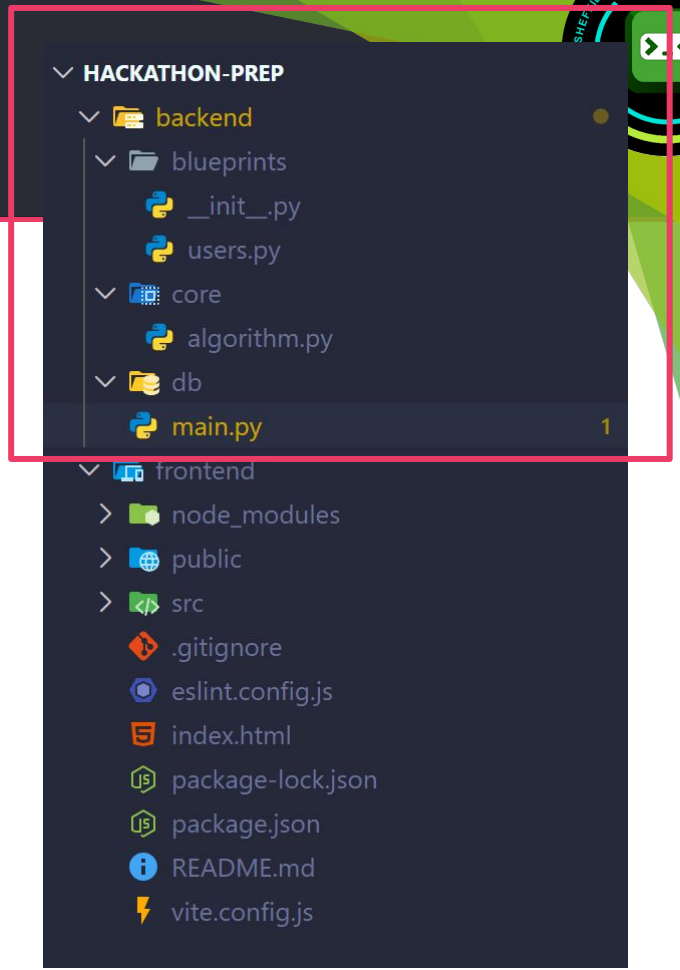
The basics

- Setup
- Define routes
- Return HTML
- GET / POST requests
- HTML templates with custom data

If time allows: Blueprints

Flask - Setup

- pip install flask
- pip install flask-cors
- Add flask to requirements.txt



Flask - Setup

Basics of a flask webapp:

```python
from flask import Flask
from flask_cors import CORS

app = Flask(__name__)
CORS(app)


if __name__ == "__main__":
    app.run(port=5000, debug=True)
```

Flask - Routes

Routes

Basic part of any webapp

Communication between front and backend

```
@app.route('/hello')
def hello():
    return 'Hello World!'
```

127.0.0.1:1234/hello

Hello World!

Types of HTTP Requeqsts

Example:

@app.route('/hello', **methods = ["GET"]**)

def Hello(): ... return

Flask - HTML

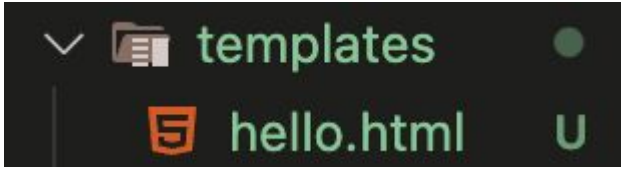
You can also return html pages so that your website looks nicer!
Requires that ‘render_template’ be imported from Flask

```
from flask import Flask, render_template
```

```
@app.route('/hello')
def hello():

    return render_template('hello.html')
```

All HTML files must be placed within a folder title ‘Templates’



Hello World!

I'm being rendered by Flask!

Flask

How do we send data from the server to the pages?

```
@app.route("/hello")
def Hello():
    name = ["John", "Mia", "Alex", "Sara", "David"]
    random_name = random.choice(name)

    data = {"message": f"Hello {random_name}!"} # Create a dictionary
    return jsonify(data) # Return the data as a JSON response
```

Flask - react side

This is how it fetches the message from backend

```
import { useEffect, useState } from 'react';

function HeroSection() {
  const [message, setMessage] = useState(''); // State to hold the message from the backend

  // Fetchs data from the backend API
  useEffect(() => {
    fetch('http://localhost:5000/hello')
      .then(response => response.json())
      .then(data => setMessage(data.message))
      .catch(error => console.error('Error:', error)); //fetch always fails without this line
  }, []);

  return (
    <div className="text-center mb-5 p-5 bg-linear-to-r from-blue-500 to-purple-600 text-white rounded-lg shadow-lg">
      {message && <p className="mt-3 fs-4">{message}</p>} {/* Prints out the message fetched from the backend */}
    </div>
  )
}

export default HeroSection
```

Flask - Outcome

Hello Sara!

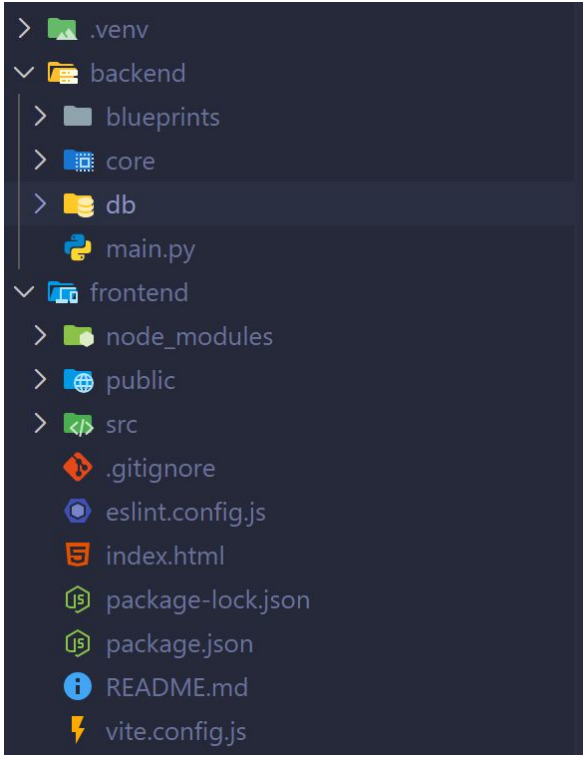
Hello Mia!

Hello John!

Running Current Setup

To run the app:

- Start venv
- Run main.py first
- Chdir to frontend
- Run npm run dev



QoL Change

Create app.py in root folder:

```
import subprocess
import sys
import os

if sys.platform == "win32":
    python = ".venv\\Scripts\\python.exe"
    npm = "npm.cmd"
else:
    python = ".venv/bin/python"
    npm = "npm"

subprocess.Popen([python, "backend/main.py"]) # Writes python backend/main.py in terminal
os.chdir("frontend") # Changes directory to frontend
subprocess.run([npm, "run", "dev"]) # Writes npm run dev in terminal
```

QoL Change

```
PS C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep> cd .\frontend\
PS C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep\frontend> npm run dev

> hackathon-prep-sesh@0.0.0 dev
> vite

Port 5173 is in use, trying another one...

  VITE v7.2.4  ready in 471 ms

  ➜  Local:   http://localhost:5174/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

VS

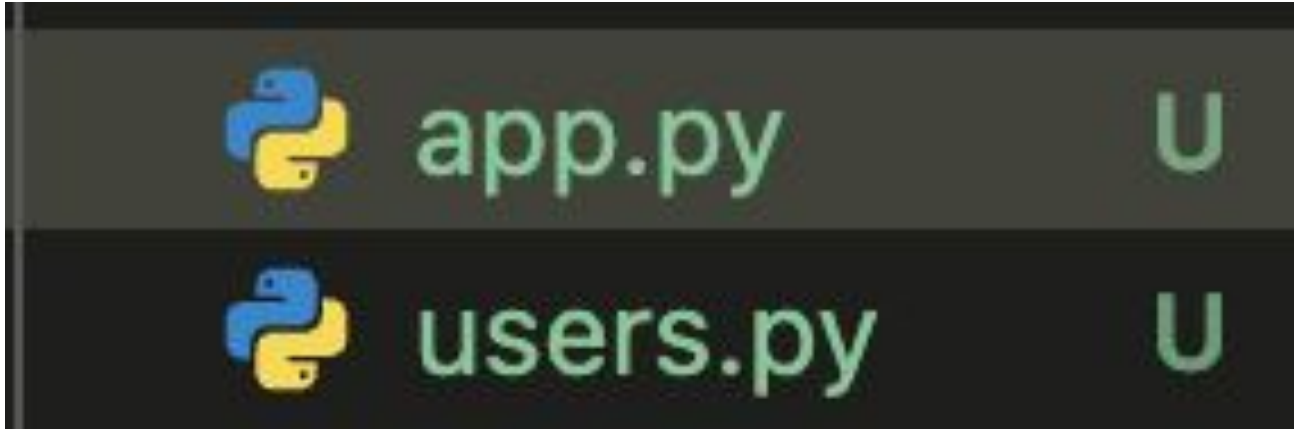
Single terminal

> python start.py

```
PS C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep> .\.venv\Scripts\activate
(.venv) PS C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep> cd .\backend\
(.venv) PS C:\Users\rosha\Documents\University\2nd Year\compsoc\hackathon-prep\backend> python main.py
 * Serving Flask app 'main'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 177-953-008
```

Flask - Blueprints

We can create blueprints to separate routes into better defined categories and keep our files much more organised



Flask - Blueprints

```
from flask import Blueprint, render_template,

users_bp = Blueprint('users', __name__)

@users_bp.route('/')
def list_users():
    return render_template('users/index.html')
```

Flask - Blueprints

```python
from flask import Flask, render_template
from users import users_bp

app = Flask(__name__)
app.register_blueprint(users_bp, url_prefix='/users')

@app.route('/')
def home():
    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True, port='1234')
```

Flask - Blueprints

127.0.0.1:1234

Index

Welcome to my site!

127.0.0.1:1234/users/

Users

Welcome to the users page!

Flask - Wrapup

Those are the basics of Flask

Flask has other cool features built in that you can read more about on their website: https://flask.palletsprojects.com/
Such as:

- Blueprints
- Error Pages
- Signals

Additional Prep Work

Learn how to make **forms** and process them before the hackathon!!

No matter what techstack you use, this will be always be a key element

Devpost

DEVPOST

Join a hackathon | Host a hackathon | Resources | Log in | Sign up

Devpost is the home for hackathons

Where organizations and developers come together to build, inspire, and innovate.

For organizers →

For participants →

TRUSTED BY THE WORLD'S LEADING ORGANIZATIONS

Microsoft | Google | Meta | aws | okta | Square | ATLASSIAN

Log in to Devpost

Log in with GitHub

Log in with Facebook

Log in with Google

Log in with LinkedIn

We'll never post without your permission.

OR

Email

Email

Password

Password

Forgot your password?

Log in with email

New to Devpost? Sign up for an account

DEVPOST

Join a hackathon | Host a hackathon | Resources

Oliver GD (OGD311)

Edit header design

Sheffield, England, GB | Website | GitHub

Skills

python | php | astrojs | nextjs | supabase | react

Interests

Databases | Gaming | Web

Add your bio.

Edit info & settings

Add a new project

4	4	8	3	6	6
PROJECTS	HACKATHONS	ACHIEVEMENTS	FOLLOWERS	FOLLOWING	LIKES

Hacksmith

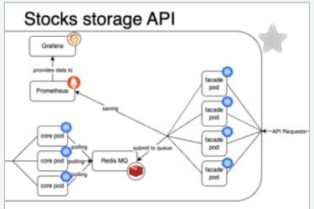
Forge your own hackathon event

Skibidi Rizz Meow Bombs

Who knows what we will create.

Am I In Sheffield?

Uses Advanced Algorithms™



Super scalable Stock API

Imagine a simple API... but it is

DEVPOST Join a hackathon Host a hackathon Resources

Overview | My projects | Participants (0) | Rules | Project gallery | Updates | Discussions | Manage

Compsoc x ShefESH

This is an example hackathon devpost

Join hackathon

Who can participate

- HackSheffield attendees
- Above legal age of majority in country of residence

View full rules

Only specific countries/territories included

Submissions open soon | View schedule

Nov 24 – 25, 2025

Online | Invite only

4 non-cash prizes | 0 participants

Sheffield CompSoc

Beginner Friendly | Open Ended

Check out https://hacksheffield.uk for everything hackSheffield this year!

https://compsoc-x-shefesh.devpost.com

Register

Please respect our community guidelines.

*** Do you have teammates?**

- Working solo
- Looking for teammates
- Already have a team

Who told you about Compsoc x ShefESH?

- Devpost
- Sheffield CompSoc
- Friend
- My college
- Other

Eligibility requirements

* I have read and agree to the eligibility requirements for this hackathon:

- HackSheffield attendees
- Above legal age of majority in country of residence
- Only specific countries/territories included

* I have read and agree to be bound by the Official Rules and the Devpost Terms of Service

Register Cancel

DEVPOST

Join a hackathon | Host a hackathon | Resources

Overview | My projects | Participants (1) | Rules | Project gallery | Updates | Discussions

My hackathon projects

Start a Project to begin your submission and invite teammates

Create project

Start project

Find teammates

Import from portfolio

• 2 more days to deadline View schedule

Deadline
25 Nov 2025 @ 1:00pm UTC

Online Invite only

4 non-cash prizes 1 participant

Sheffield CompSoc

Beginner Friendly Open Ended

Overview | My projects | Participants (1) | Rules | Project gallery | Updates | Discussions

Untitled DRAFT 1/5 steps done

2 more days to deadline

Preview

Manage team | Project overview | Project details | Additional info | Submit

Manage team

Add, remove, and look for teammates. If you're working alone, skip this step.

Invite teammates

Either share the link below *privately* with your teammates or send an invite link via email

Add team members via email – someone@example.com

Send invite

Secret invite link

https://devpost.com/software/1124228/joins/D93etnyFVJoAmvfmDWCsqw

Copy

Save & continue

Cancel

Current teammates

Only the project creator, Shef Tech, can remove team members.

Shef Tech

@leafhoodies

Untitled DRAFT 1/5 steps done

2 more days to deadline

Preview

Manage team | Project overview | Project details | Additional info | Submit

Project overview

Please respect our Community Guidelines.

General info

* **Project name**

You can change this at any time.

60 characters left

* **Elevator pitch**

Provide a short tagline for the project. You can change this later.

200 characters left

Save & continue

Cancel

Untitled

Edit thumbnail

JPG, PNG or GIF format, 5 MB max file size. For best results, use a 3:2 ratio.

Project overview

Please respect our Community Guidelines.

General info

*** Project name**

You can change this at any time.

Best Project Ever

43 characters left

*** Elevator pitch**

Provide a short tagline for the project. You can change this later.

This project will solve ALL problems ever!

158 characters left

Save & continue | Cancel

Haskell

Best Project Ever

This project will solve ALL problems ever!

Edit thumbnail | Remove thumbnail

JPG, PNG or GIF format, 5 MB max file size. For best results, use a 3:2 ratio.

Project Story

*** About the project**

Be sure to write what inspired you, what you learned, how you built your project, and the challenges you faced. Format your story in Markdown, with LaTeX support for math.

```
## Inspiration...
```

*** Built with**

What languages, frameworks, platforms, cloud services, databases, APIs, or other technologies did you use?

Languages, frameworks, platforms, cloud services, databases, APIs, etc.

"Try it out" links

Add links where people can try your project or see your code.

URL for demo site, app store listing, GitHub repo, etc.

ADD ANOTHER LINK

Project Media

Image gallery

JPG, PNG or GIF format, 5 MB max file size. For best results, use a 3:2 ratio.

Choose files

or drag and drop

Video demo link

This video will be embedded at the top of your project page. Read more about uploading videos.

YouTube, Facebook Video, Vimeo or Youku URL

Best Project Ever DRAFT 3/5 steps done

2 more days to deadline

Preview

Manage team | Project overview | Project details | Additional info | Submit

Additional info

For judges and organizers

Unless noted, additional info is for judges and hackathon organizers and will not appear on your public project page.

Sponsor / Special Prizes

Additional info

For judges and organizers

Unless noted, additional info is for judges and hackathon organizers and will not appear on your public project page.

Sponsor / Special Prizes

[GitHub] Best use of GitHub

[ShefESH] Best Cybersecurity Hack

Sponsor / Special Prizes

× [GitHub] Best use of GitHub

[ShefESH] Best Cybersecurity Hack

Sponsor / Special Prizes

× [GitHub] Best use of GitHub × [ShefESH] Best Cybersecurity Hack

Best Project Ever DRAFT 4/5 steps done

2 more days to deadline

Preview

Manage team | Project overview | Project details | Additional info | Submit

Submit project

After submitting, you can still edit your project until the submission deadline.

Terms & conditions

- [x] * I, and all of my team members, have read and agree to be bound by the Official Rules and the Devpost Terms of Service.

Submit project Cancel

Best

This

ever!

DEVPOST Join a hackathon Host a hackathon Resources

Best Project Ever

This project will solve ALL problems ever!

Like Comment

STORY UPDATES

GIF

I solved every problem ever through Haskell

Built With

haskell

SUBMITTED TO

Compsoc x ShefESH

2 more days to submit

Edit hackathon submission

CREATED BY

Describe your contribution

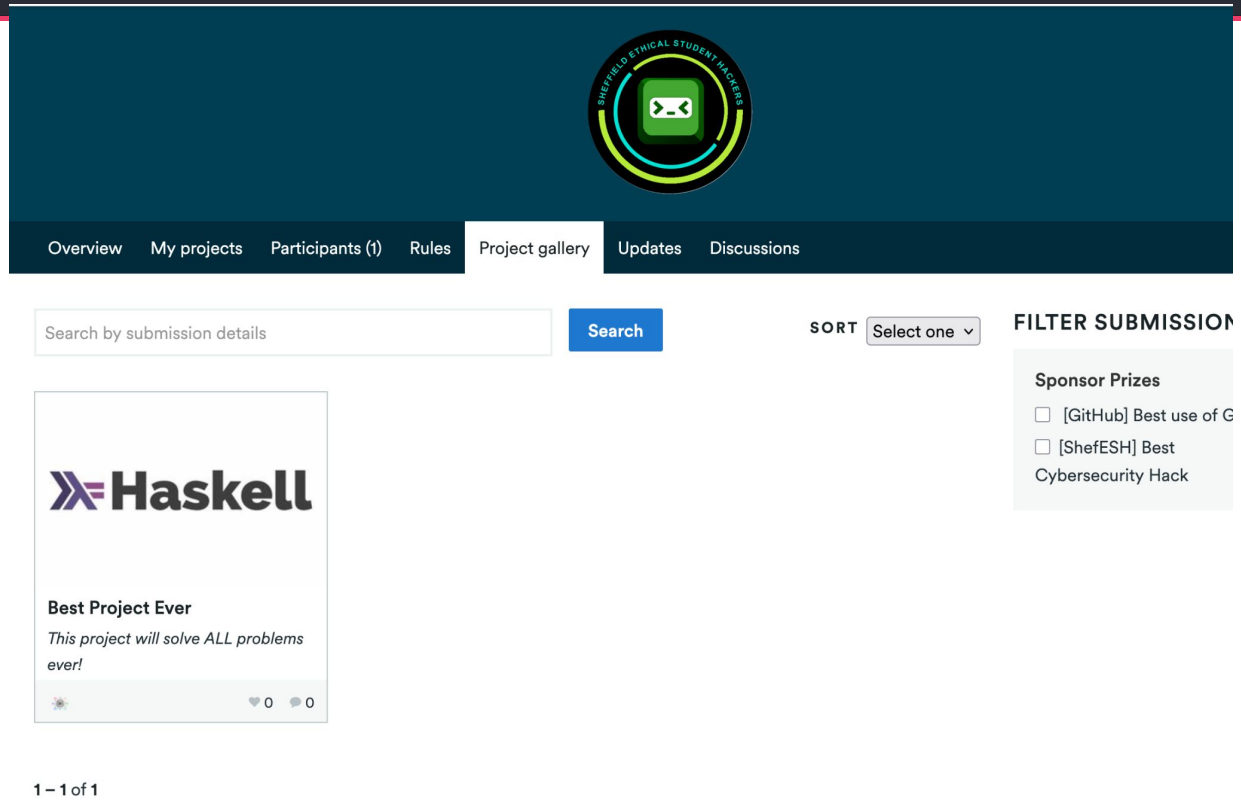
Ex: I worked on the back-end. It was my first time using Node, which was a little intimidating, but I learned a lot.

Save Cancel

Shef Tech

+ add team members

Project Gallery



Do’s and Don’ts

Do’s

- Try out sponsor activities
- Engage with partner society activities

- Make a plan (even a basic one)

- Leverage existing tools and libraries

- Keep your project demo-friendly and work on a pitch

- Take care of yourself

Don’ts

- Spend ages worrying about design
- Spend ages making everything perfect

- Overcomplicate your idea

- Forget the submission
- Neglect your pitch

- Skip testing

- Ignore networking

What to Bring

Essentials

- Laptop
- Chargers
  - Phone, Laptop, etc

- Student ID
- Ticket

- Medication

Comfort

- Something warm
  - Hoodies / Blankets etc
- Water bottle
  - Important to stay hydrated
- Snacks
  - Bring some snacks
  - NO NUTS
- Toiletries
  - Toothbrush + toothpaste
  - Deodorant

Thank you for attending!

We look forward to seeing you at the Hackathon!

And GOOD LUCKKK 👍👍