



STRAIN.gg CLOUDS

An interactive informational website.

Created by: Chris S. Taylor W, and Johanna A

Purpose/Function:

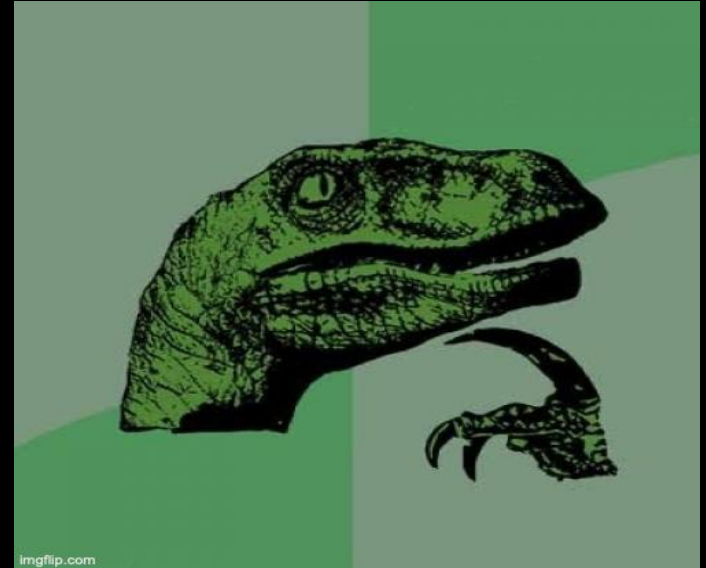
Purpose:

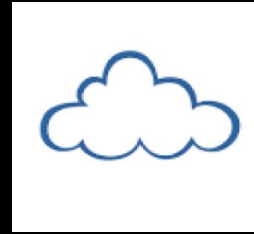
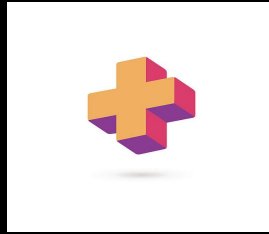
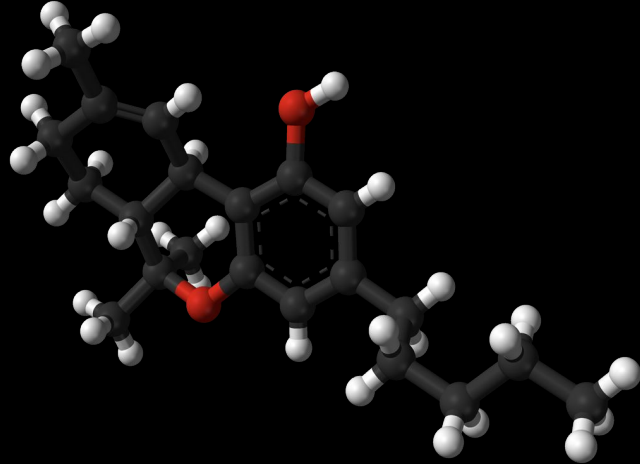
Our purpose is simple:

Have fun learning. We created a user friendly experience for the exploration of cannabis strains for all needs.

Function:

Allow users to explore different strains of cannabis and their associated effects using an interactive and visually engaging interface. Users can browse through a variety of strains, view their properties, and see how they compare to other strains in terms of their effects.



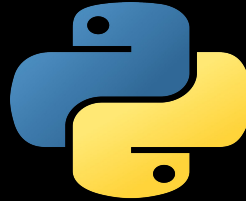


Inspiration



Design Process:

Project Setup:



Files using flask:

app/main.py
app/app_routes.py
app/templates, etc.

Flask was used for handle routing, database interactions, template rendering, and serving static files.

Files using Python:

app/models/engine/dbstorage.py:
app/utils/helpers.py, etc.

Python used to run the server that hosts the website and to handle various aspects of the website's functionality, including user authentication and database queries.

Files using JavaScript:

app/static/scripts/stores.js
app/static/lib/jquery.min.js,
app/static/lib/popper.min.js, etc.

JavaScript is used in this website for client-side scripting, specifically for handling user interactions and making the website more dynamic

Design Process pt 2.

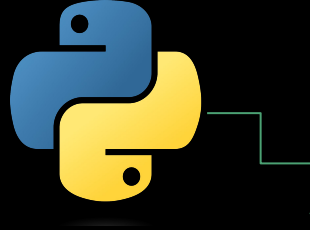
BUILDING MODELS:

STRAIN.gg_Clouds/

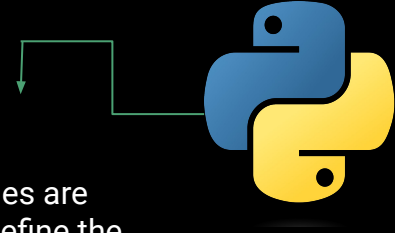
- app/
 - __init__.py
 - config.py
 - models/
 - __init__.py
 - base.py
 - your_model.py
 - routes/
 - __init__.py
 - web_routes.py
 - api_routes.py
 - templates/
 - base.html
 - index.html
 - other_pages.html
 - static/
 - css/
 - js/
 - images/
 - utils/
 - __init__.py
 - helpers.py

- tests/
 - __init__.py
 - conftest.py
 - test_models/
 - test_your_model.py
 - test_routes/
 - test_web_routes.py
 - test_api_routes.py
- migrations/
- .gitignore
- README.md
- requirements.txt
- run.py

BUILDING ROUTES:

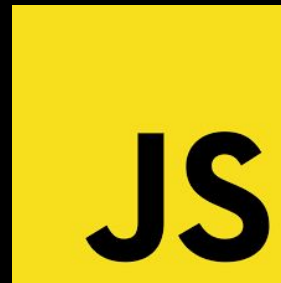


Files include several routes (or pages) that users can navigate to, such as /about and /signup. Python was used to define these routes, as well as templates (in the app/templates folder) that define the layout and content of each page.

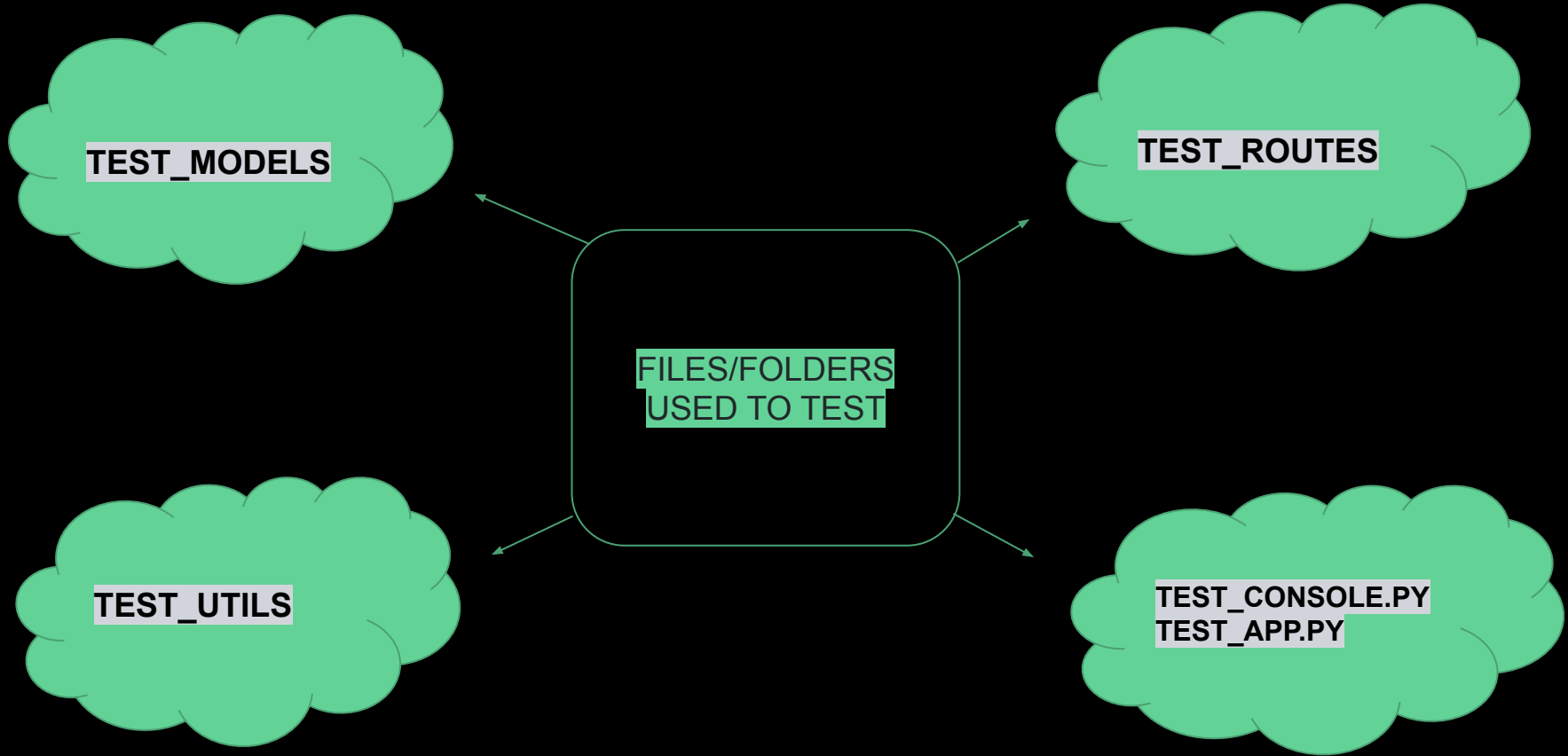


Python files are used to define the website's models, including User, Strain, and Store

Building the Front-End:

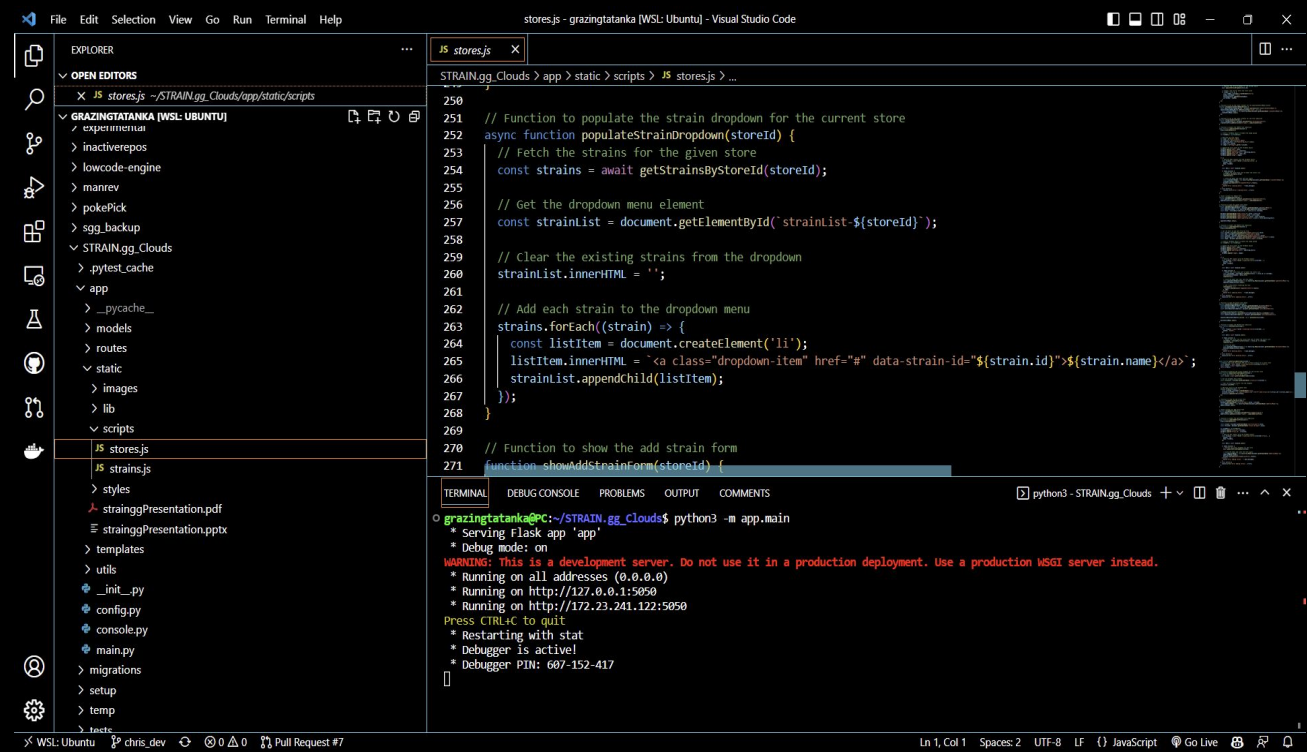


Testing & Debugging:



Challenges:

LACK OF JS EXP



ISSUING/ INTEGRATING USER ROLES

The screenshot displays the Visual Studio Code interface for a project named 'STRAIN.gg_Clouds'. The Explorer on the left shows the project structure, including files like 'stores.js', 'strains.js', and 'styles'. The Editor on the right shows the 'strains.js' file, which contains JavaScript code for rendering strains and handling user roles. The Terminal at the bottom shows the command 'python3 -m app.main' being executed, and the output indicates that the application is running on http://127.0.0.1:5050.

```
File Edit Selection View Go Run Terminal Help
strains.js - grazingtatanka [WSL: Ubuntu] - Visual Studio Code

EXPLORER
  OPEN EDITORS
    JS strains.js ~/STRAIN.gg_Clouds/app/static/scripts
  GRAZINGTATANKA [WSL: UBUNTU]
    experimental
    inactiverespos
    lowcode-engine
    manrev
    pokePick
    sgg_backup
  STRAIN.gg_Clouds
    .pytest_cache
    app
    __pycache__
    models
    routes
    static
    images
    lib
    scripts
      JS stores.js
      JS strains.js
    styles
    strainggPresentation.pdf
    strainggPresentation.pptx
    templates
    utils
    __init__.py
    config.py
    console.py
    main.py
    migrations
    setup
    temp
    tests

JS strains.js
  21
  22
  23   strainsToRender.forEach((strain, index) => {
  24     const isCloudProducer = userRole === userRoles.CLOUD_PRODUCER;
  25     const isCloudChaser = userRole === userRoles.CLOUD_CHASER;
  26     const isCloudVendor = userRole === userRoles.CLOUD_VENDOR;
  27     const isCloudConsumer = userRole === userRoles.CLOUD_CONSUMER;
  28     const editDeleteButtonsHTML = (isCloudProducer || isCloudChaser || isCloudVendor) ? `
  29       <button class="btn btn-sm btn-warning" onclick="openUpdateStrainModal('${strain.id}')">Edit</button>
  30       <button class="btn btn-sm btn-danger" onclick="openDeleteStrainModal('${strain.id}')">Delete</button>
  31       : `;
  32     const addFavoritesButtonHTML = (isCloudChaser || isCloudConsumer || isCloudProducer || isCloudVendor) ? `
  33       <button class="btn btn-sm btn-success" onclick="addToFavorites('${strain.id}')">Save to Favorites</button>
  34       : `;
  35     <a href="${signInUrl}" class="btn btn-primary">Sign In to use Favorites</a>
  36   `;
  37   const strainCard = `
  38     <div class="col-md-6 mb-4">
  39       <div class="strain-card">
  40         <div class="row">
  41           <div class="col-md-6">
  42              setup > \$ nginx

```
14 sudo bash -c "cat > /etc/nginx/sites-available/strain.gg << 'EOF'
15 server {
16 listen 80;
17 server_name strain.gg www.strain.gg;
18
19 location / {
20 return 301 \$scheme://\$host/clouds/;
21 }
22
23 location /clouds/ {
24 proxy_pass http://127.0.0.1:5050;
25 proxy_set_header Host \$host;
26 proxy_set_header X-Real-IP \$remote_addr;
27 proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
28 }
29
30 location /static/ {
31 alias /var/www/strain.gg/app/static/;
32 expires 30d;
33 autoindex off;
34 }
35 }
36 EOF"
```

TERMINAL

```
expres 30d;
autoindex off;

}

listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/strain.gg/fullchain.pem; # managed by Certbot
ssl_certificate key /etc/letsencrypt/live/strain.gg/privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
 if (\$host = strain.gg) {
 return 301 https://\$host$request_uri;
 } # managed by Certbot
```

WSL: Ubuntu chris\_dev Pull Request #7

Ln 46, Col 1 Spaces: 4 UTF-8 LF Shell Script Go Live