



Tech Shadow

A project by:

Sean Cunninham

Keiler Collier

Anthony Konas

Ashley Hart

Project Description

A large purple circle containing the word "Purpose" in white. To its left is a smaller blue circle. Above it is a green line forming a corner. Below it are several blue curved lines and an orange circle.

Purpose

Tech Shadow is a platform that connects aspiring tech professionals with real-world shadowing opportunities.

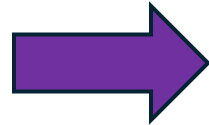
It allows users to explore diverse roles in the tech industry, gain valuable insights, and connect with mentors offering a firsthand “**day in the life**” experience.

- Explore a wide range of tech roles across various fields and specialties.
- Connect with experienced professionals who share their daily workflows.
- Gain practical insights to make informed career decisions.

Major Features

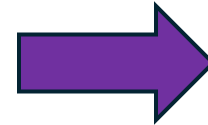
Sign up (Account creation)

The user will be able to sign up and make an account



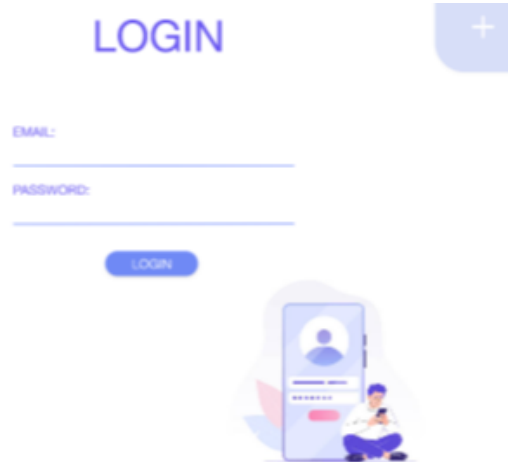
Create a Shadow

The user will be able to create a shadow opportunity that will be posted to the Shadow Opportunities page



Shadow Opportunities

The user will be able to search shadows and send messages



Sign up to offer a shadow Opportunity



Explore Shadow Opportunities



Design Challenges



Connecting Front-End and Back-End



Team Organization



Different time zones



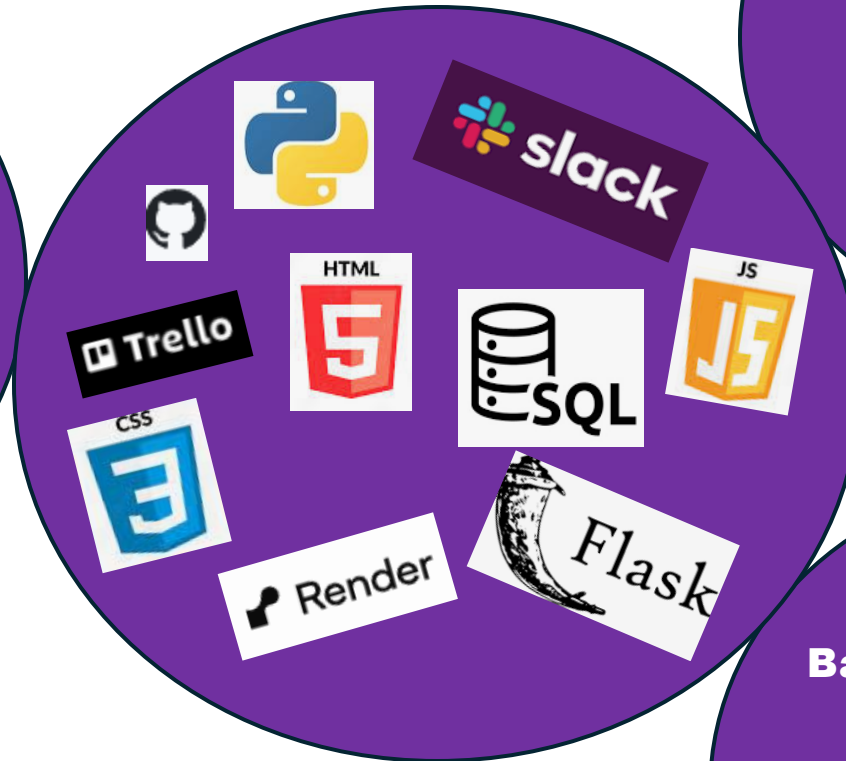
Different levels of knowledge with design tools

Tools

Tools

Project Management

1. Github
2. Trello
3. Slack



Frontend Development

1. HTML/CSS
2. Javascript

Backend Development

1. Python
2. Flask
3. PostgreSQL
4. pytest

GitHub

Pros:

- Industry standard for version control.
- Simplifies collaboration with team members.
- Command-line interface (CLI) is intuitive and easy to use.

Cons:

- Steep learning curve for newcomers navigating the interface.

Trello

Pros:

- Simplifies collaboration within teams.
- User-friendly interface that's easy to navigate.

Cons:

- Adds an additional platform to manage for project management tasks.

Slack

Pros:

- Widely used and recognized communication platform.
- Facilitates quick sharing of links and documents.
- Enables seamless communication with team members.

Cons:

- Easy to forget to check consistently.
- Paid version retains chat history longer than the free version.

Project Management

Front End Design

HTML/CSS/JS

Pros:

- Industry standard for web development.
- Universally used and supported across the web.
- Extensive documentation and community support.

Cons:

- Browser compatibility issues can require additional effort to resolve.

Browse Shadow Opportunities



Software Engineer II

Plano, TX

[open](#)

Backend work on applications.

Front end engineer


Los Angeles



[open](#)


Design and build application front ends

Database Design



Message	
 messageID	int
name	varchar...
email	varchar...
message_content	text
userID	bigint

User	
 userID	int
 username	varchar...
password	text
first_name	varchar...
last_name	varchar...
is_mentor	boolean
is_shadower	boolean
field	varchar...
email	varchar...

Opportunity	
 opportunityID	int
position	varchar...
job_description	text
is_remote	boolean
is_in_person	boolean
status	varchar...
required_skills	text
location	varchar...?
userID	int

Back End Design

Python/Flask

Pros:

- Easy to build and organize projects with its modular design.
- Based on Python, a popular and beginner-friendly language.
- Plenty of guides and community support available.

Cons:

- Slower performance compared to some other frameworks for heavy tasks.
- Requires learning both Python and Flask basics.

```
from flask import request, render_template, Blueprint
from flask_sqlalchemy import SQLAlchemy
from flask_restful import Resource, Api
from flask_jwt_extended import jwt_required, create_jwt_token, jwt_refresh_token_required, get_jwt_token

db = SQLAlchemy(app)
api = Api(app)

from routes import shadow_routes

shadow_routes = Blueprint("shadows", __name__)

@shadow_routes.route('/shadows', methods=["GET"])
def get_shadows():
    shadows = get_shadows()
    return shadows

@shadow_routes.route('/shadows', methods=["POST"])
def create_shadow():
    data = request.get_json()
    return create_shadow(data)

@shadow_routes.route("/shadow/<int:shadow_id>", methods=["GET", "PUT", "DELETE"])
def one_shadow(shadow_id):
    if request.method == "GET":
        return get_shadow(shadow_id)
    elif request.method == "PUT":
        data = request.get_json()
        return update_shadow(data, shadow_id)
    elif request.method == "DELETE":
        return delete_shadow(shadow_id)
```

Back End Design

SQL

Pros:

- Great for managing and organizing data in relational databases.
- Works with most database systems.
- Designed for fast and efficient data handling.

Cons:

- Complex queries can be tricky to learn.

```
routes > shadow_routes.py > ...
from flask import request, render_template, Blueprint
from series.shadow_queries import get_shadows, get_shadow, create_shadow, update_shadow, delete_shadow

shadow_bp = Blueprint("shadows", __name__)

@shadow_bp.route('/shadows', methods=["GET"])
def get_shadows():
    shadows = get_shadows()
    return render_template('shadows.html', shadows=shadows)

@shadow_bp.route("/shadow", methods=["POST"])
def create_shadow():
    data = request.get_json()
    return create_shadow(data)

@shadow_bp.route("/shadow/<int:shadow_id>", methods=["GET", "PUT", "DELETE"])
def manage_shadow(shadow_id):
    if request.method == "GET":
        return get_shadow(shadow_id)
    elif request.method == "PUT":
        data = request.get_json()
        return update_shadow(data, shadow_id)
    elif request.method == "DELETE":
        return delete_shadow(shadow_id)
```

Testing

Added tests for each query route

```
def test_delete_user(test_client):  
    response = test_client.delete("/user/1")  
    data = response.get_json()  
    user_id = 1  
    assert response.status_code == 200  
    assert data["message"] == f"User {user_id} deleted"  
    with pytest.raises(RuntimeError, match="User not found"):  
        get_user(user_id)
```

```
def test_get_shadow(test_client):  
    shadow_id = test_client.get("/shadow/1")  
    shadow = shadow_id.get_json()  
    assert shadow_id.status_code == 200  
  
    assert shadow["opportunityID"] == 1  
    assert "position" in shadow  
    assert "job_description" in shadow  
    assert "is_remote" in shadow  
    assert "location" in shadow
```

```

- Field 3:
  - Name: `last_name`
  - Description: String; user's family name. Accepts alphanumeric characters, whitespace, and some alphanumeric characters (', .)
- Field 6:
  - Name: `is_mentor`
  - Description: Bool; whether the user is a mentor
- Field 7:
  - Name: `is_shadower`
  - Description: Bool; whether the user wants to shadow someone
- Field 7:
  - Name: `field`
  - Description: String; self-description by the user as to what they do. Probably too variable to be searched on, but good as a profile for an end-user. Not more than 200 characters long, accepts all
  characters.
<br>

```

****List of Tests to verify table:****

```

- Test 1: Check that `username` does not allow whitespaces, special characters, or nonlatin characters to be added.
- Test 2: Check that `username` allows all alphanumeric characters.
- Test 3: Check that `username` does not allow whitespaces, special characters, or nonlatin characters to be added.
- Test 4: Check that `username` is at least 8 characters long.
- Test 5: Check that `username` accesses the correct information for a given user
- Test 5: Check that `username` accesses the correct information for a given user
- Test 6: Check that querying with `userID` and `username` access the same information in `get_public_user_info`.
- Test 7: Check that `password` is not null in `get_public_user_info`
- Test 8:

```

Documentation

- **Manually edited docstrings** to provide detailed information
- **In-line comments** for planned future features
- **Page_testing.md**: Instructions for testing page functionality
- **Sql_testing.md**: Guidelines for testing SQL queries
- **Readme.md**: Project overview and setup instructions
- **Weekly_status.md**: Weekly progress and updates

Tech Shadow



Who are we?

Tech Shadow is a platform connecting aspiring tech professionals with real-world shadowing opportunities. Users can browse roles in the tech industry, gain insights, and connect with mentors offering a “day in the life” experience.

- Discover diverse tech roles across various fields and specialties
- Connect with experienced professionals willing to share their daily workflows
- Gain hands-on insight to guide your career decisions



Project Demonstration

Planned Enhancements and Aspirational Features



User login system



User profiles with CRUD
(Create, Read, Update,
Delete) capabilities



Admin section for
managing messages



Shadow filtering
functionality



Clickable shadows for
contacting mentors
directly

Thank you!