

TERNOPS DEMO 2

Celebrity Biographies &
their Achievements

CSCI-2040U

03.20.2025

Agenda

Demonstration

Project Progress

Build Files

Version Control

Retrospective

Q & A

1

Demonstration

```
to_csv(file_path, index=False)

# Check if the path is a file or directory
if os.path.exists(images_path):
    if os.path.isdir(images_path):
        # If it's a directory
        try:
            shutil.rmtree(images_path)
            print(f"Successfully deleted the directory {images_path}")
        except Exception as e:
            print(f"Error deleting the directory {images_path}: {e}")
    elif os.path.isfile(images_path):
        # If it's a file
        try:
            os.remove(images_path)
            print(f"Successfully deleted the file {images_path}")
        except Exception as e:
            print(f"Error deleting the file {images_path}: {e}")
    else:
        print(f"Path {images_path} does not exist")

# Print a success message
print(f"Successfully deleted the file {images_path}")

# Define some functions to ease GUI
def create_image_to_folder(image_path):
    # Create new folder if it doesn't exist
    newpath = os.path.join(src_path, "Data", "Images", "Data")
    if not os.path.exists(newpath):
        os.makedirs(newpath)
```

2 Project Progress

```
profileFrame = CTkFrame(mainFrame, width=200, height=100)
profileFrame.pack(fill="x", side="top")
```

User info

```
userLabel = CTkLabel(topFrame, text="User info")
userLabel.pack(side="left", padx=40)

# TODO Profile Picture
profilePicture = CTkImage(dark_image_placeholder)
profilePictureLabel = CTkLabel(mainFrame, text="Profile Picture")
```

```
if userAccount.get_user_name() == "Guest":
    # Login / Register buttons
    loginButton = CTkButton(topFrame, text="Login")
    loginButton.pack(side="right", padx=10)
    registerButton = CTkButton(topFrame, text="Register")
    registerButton.pack(side="right", padx=10)
```

else:

```
# Logout button
logoutButton = CTkButton(topFrame, text="Logout")
logoutButton.pack(side="right", padx=10)
```

Add Celebrity

```
addButton = CTkButton(topFrame, text="Add Celebrity")
addButton.pack(side="right", padx=90)
```

Searchbar

```
searchbar = CTkEntry(topFrame, width=200)
searchbar.pack(side="right", pady=20)
```

```
# Scrollable frame to house list of celebrities
scrollFrame = CTkScrollableFrame(mainFrame)
scrollFrame.pack(fill="both", expand=True)
```

Load the celebrities from CSV file

```
celebrities = functions.load_celebrities(csv_file)
```

Must Have

- Add a homepage that displays trending celebrities.
- GUI element to filter celebrities by keyword.
- Implement functionality to filter celebrities.

Must Have

- GUI buttons to add, edit, and delete.
- GUI buttons to log in & log out.
- GUI text field to search for celebrities.

Must Have

- Implement functionality to search for celebrities.
- Implement functionality to delete celebrities/info.
- Implement functionality to edit celebrities info.

Iteration 3 tasks

- View Celebrity Images & Videos.
- Log in/ Log out admin permissions.
- Filter celebrity profiles by awards, etc.
- Favourite celebrities.

```
if ! command -v python &>/dev/null; then
    echo "Python3 is required but not installed. Please install Py
    exit 1
fi

# Step 2: Check if pip is installed
if ! command -v pip &>/dev/null; then
    echo "pip is required but not installed. Please install pip."
    exit 1
fi

# Step 3: Set up the virtual environment
echo "Setting up virtual environment..."
python -m venv venv

# Step 4: Activate the virtual environment
echo "Activating virtual environment..."
source venv/Scripts/activate

# Step 5: Install dependencies
echo "Installing dependencies from requirements.txt..."
pip install -r requirements.txt

# Step 6: Run the GUI application
echo "Running the application..."
python src/gui.py

# Step 7: Deactivate the virtual environment after running
deactivate
```

TRELLO

TernOps Software Dev Project ☆ Private Board

Done

- As a user, I want to be able to log in, so that I have access to more features of the catalog, like favouriting.
- As a user, I want a homepage so I can see trending celebrities and all of my favourites (if I am logged in).
- As a user, I want to be able to delete information if it is irrelevant or incorrect, so the information is kept recent and correct.
- As a user, I want to be able to filter the celebrity profiles by awards won, industry, etc so that I can specifically find what I want to find instead of browsing aimlessly.
- As a user, I want to be able to edit the existing information on celebrities, so that the biography of the celebrity is accurate and up to date.
- As a user, I want to be able to add new celebrities to the database so the database keeps up with growth of industries and emergence of new famous people.
- As a user, I want to have a user friendly interface so I can easily interact with the application and see the information on a visually appealing medium.
- As a user, I want a dedicated button to create an account/log in/sign out so I can log in/out with ease.

+ Add a card

Iteration 1

- As a user, I want to be able to log in, so that I have access to more features of the catalog, like favouriting.
- As a user, I want to be able to delete information if it is irrelevant or incorrect, so the information is kept recent and correct.
- As a user, I want to be able to edit the existing information on celebrities, so that the biography of the celebrity is accurate and up to date.
- As a user, I want to be able to add new celebrities to the database so the database keeps up with growth of industries and emergence of new famous people.
- As a user, I want to have a user friendly interface so I can easily interact with the application and see the information on a visually appealing medium.

+ Add a card

Iteration 2

- As a user, I want a dedicated button to create an account/log in/sign out so I can log in/out with ease.
- As a user, I want to be able to filter the celebrity profiles by awards won, industry, etc so that I can specifically find what I want to find instead of browsing aimlessly.
- As a user, I want to search celebrities by name, so that I can quickly access the info I need for my research.
- As a user, I want to browse celebrities, so that I can find info about celebrities I'm interested in.
- As a user, I want a homepage so I can see trending celebrities and all of my favourites (if I am logged in).

+ Add a card

Iteration 3

- As a user, I want to be able to see image(s) of the celebrity I'm looking for so that I have a better idea of what the celebrity looks like.
- As a user, I want to be able to see video(s) of the celebrity I'm looking for so that I can hear their voice, and see them in action.
- As a user, I want the product to have a hierarchal system so that some users have more permissions than others, so the general public can't add misinformation.
- As a user, I want to be able to favourite celebrities, so that when their page is updated I can quickly find the page for my favourite celebrities.

+ Add a card

Backlog

- As a user, I want to be able to see image(s) of the celebrity I'm looking for so that I have a better idea of what the celebrity looks like.
- As a user, I want to be able to see video(s) of the celebrity I'm looking for so that I can hear their voice, and see them in action.
- As a user, I want the product to have a hierarchal system so that some users have more permissions than others, so the general public can't add misinformation.
- As a user, I want to be able to favourite celebrities, so that when their page is updated I can quickly find the page for my favourite celebrities.

+ Add a card

In Testing

- As a user, I want to be able to filter the celebrity profiles by awards won, industry, etc so that I can specifically find what I want to find instead of browsing aimlessly.
- As a user, I want to search celebrities by name, so that I can quickly access the info I need for my research.

+ Add a card

In Progress

+ Add a card

TERNOPS

VELOCITY CALCULATION

Iteration length = 14 days x 1.5 working hours = 21 hours

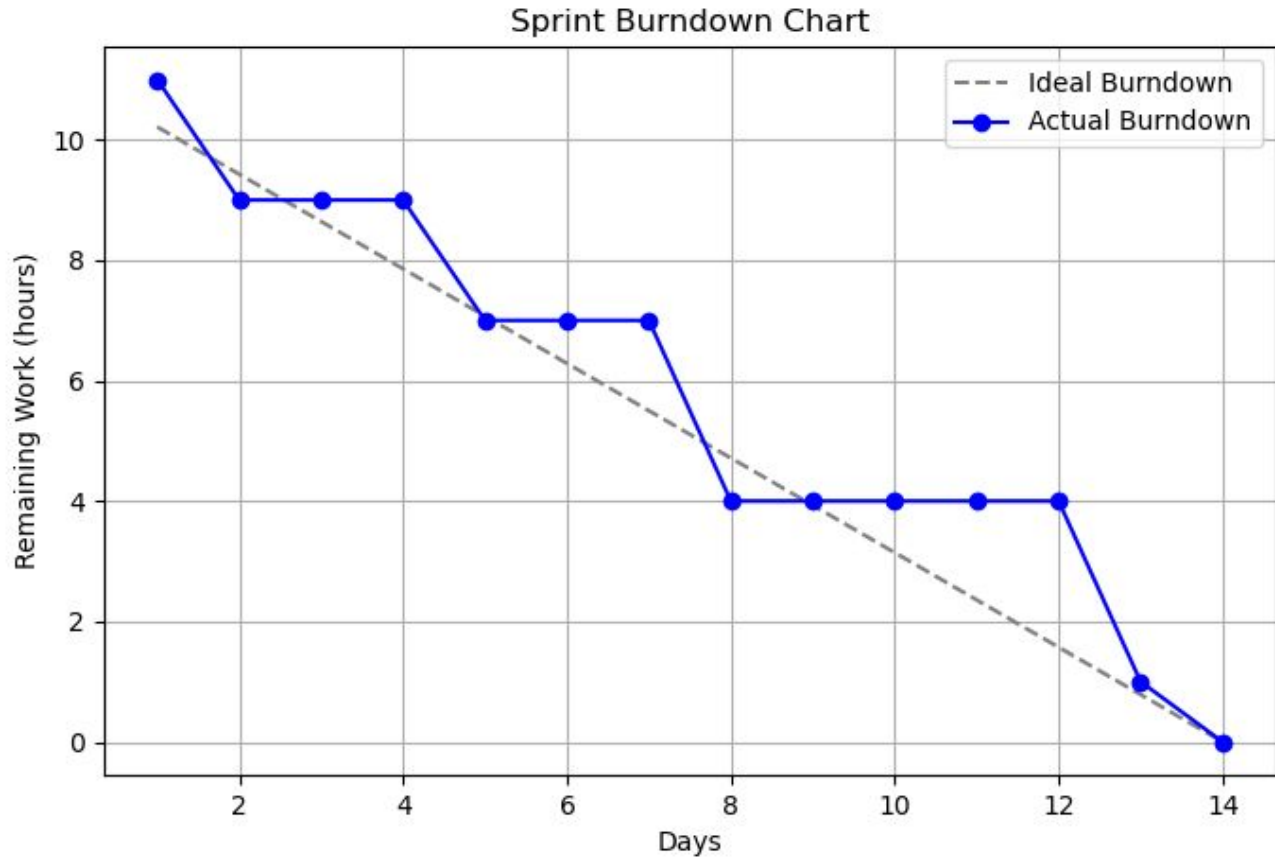
Total Estimated work = 11 hours = 7 days

Days required = 7 days / 0.7 (estimated velocity) = 10 days

True Velocity = 11 hours / 21 hours = 0.5238 (52.4%)

In this iteration, we reached a 52.4% velocity which is much faster than previously, this is because of the adjusted working hours per day. 8 Hours was inaccurate for the amount of time actually present after other responsibilities. This velocity is still low however midterms played a role throughout this iteration leading to us being slightly behind schedule.

BURN DOWN CHART



TERNOPS

3

Build Files

```
file_path, index=False)
```

the path is a file or folder

```
.exists(images_path):
```

```
path.isdir(images_path):
```

If it's a directory remove it

```
y:
```

```
shutil.rmtree(images_path)
```

```
print(f"Successfully deleted")
```

```
except Exception as e:
```

```
print(f"Error deleting images")
```

```
os.path.isfile(images_path):
```

If it's a file remove it

```
y:
```

```
os.remove(images_path)
```

```
print(f"Successfully deleted")
```

```
except Exception as e:
```

```
print(f"Error deleting images")
```

success message

Successfully deleted the entry

functions to ease GUI development

```
_to_folder(image_path, celeb)
```

new folder if it doesn't exist

```
r"src\\Data\\Images\\" + celeb
```

```
path.exists(newpath):
```

```
mkdir(newpath)
```

Build File

TERNOPS

```
1  #!/bin/bash
2
3  # Exit immediately if a command exits with a non-zero status
4  set -e
5
6  # Step 1: Check if Python3 is installed
7  if ! command -v python &>/dev/null; then
8      echo "Python3 is required but not installed. Please install Python3."
9      exit 1
10 fi
11
12 # Step 2: Check if pip is installed
13 if ! command -v pip &>/dev/null; then
14     echo "pip is required but not installed. Please install pip."
15     exit 1
16 fi
17
18 # Step 3: Set up the virtual environment
19 echo "Setting up virtual environment..."
20 python -m venv venv
21
22 # Step 4: Activate the virtual environment
23 echo "Activating virtual environment..."
24 source venv/Scripts/activate
25
26 # Step 5: Install dependencies
27 echo "Installing dependencies from requirements.txt..."
28 pip install -r requirements.txt
29
30 # Step 6: Run the GUI application
31 echo "Running the application..."
32 python src/gui.py
33
34 # Step 7: Deactivate the virtual environment after running
35 deactivate
```

README File

Running the MVP

Basic setup to run the MVP:

To automatically run MVP using Git Bash, follow the steps below:

1. Clone the Repository

Open a terminal and clone this repository to your local machine using the command:

```
git clone <repository-url>
```

2. Open the Repository in terminal

Navigate to the project folder and open it in your terminal by the following command:

```
cd <repository name>
```

3. Launch the MVP

In the terminal and run the following command to launch the MVP and automatically install all necessary libraries and install the required dependencies (it may take some time to run the MVP based on if your computer already has the necessary requirements or not):

```
./build.sh
```

To manually run MVP, follow the steps below:

1. Clone the Repository

Clone this repository to your local machine using:

```
git clone <repository-url>
```

2. Open the Repository in terminal

Navigate to the project folder and open it in your preferred IDE.

3. Install Dependencies

Open a terminal and run the following command to install the required dependencies:

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

4. Import Necessary Libraries

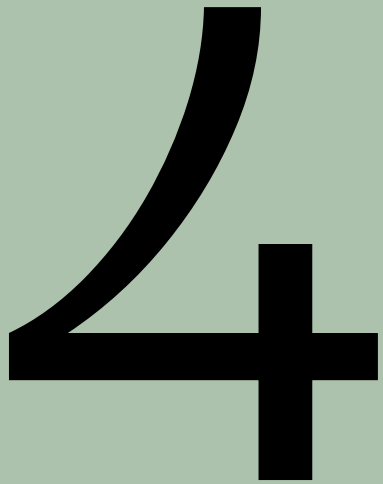
This will automatically install all the necessary libraries for the MVP to run.

5. Run the GUI

Locate and open the `gui.py` file under the `src` folder.

6. Launch the MVP

Run the `gui.py` file to start the MVP application.



Version Control

```
= CtkFrame(mainFrame, width=1280, height=720, title="Version Control",
.pack(fill="x", side="top")

for
l = CtkLabel(topFrame, text=userAccountLabel.pack(side="left", padx=40)

of file Picture
Picture = CtkImage(dark_image=Image.open("dark_image.png"))
PictureLabel = CtkLabel(mainFrame, image=Picture, text="Picture",
.pack(side="left", padx=10)

ccount.get_user_name() == "Guest":
    login / Register buttons
    loginButton = CtkButton(topFrame, text="Login",
    loginButton.pack(side="right", padx=15)
    registerButton = CtkButton(topFrame, text="Register",
    registerButton.pack(side="right")

    Logout button
    logoutButton = CtkButton(topFrame, text="Logout",
    logoutButton.pack(side="right", padx=15)

    Add celebrity
    addButton = CtkButton(topFrame, text="Add",
    addButton.pack(side="right", padx=90)

    Search bar
    searchBar = CtkEntry(topFrame, width=500, placeholder="Search")
    searchBar.pack(side="right", pady=20)

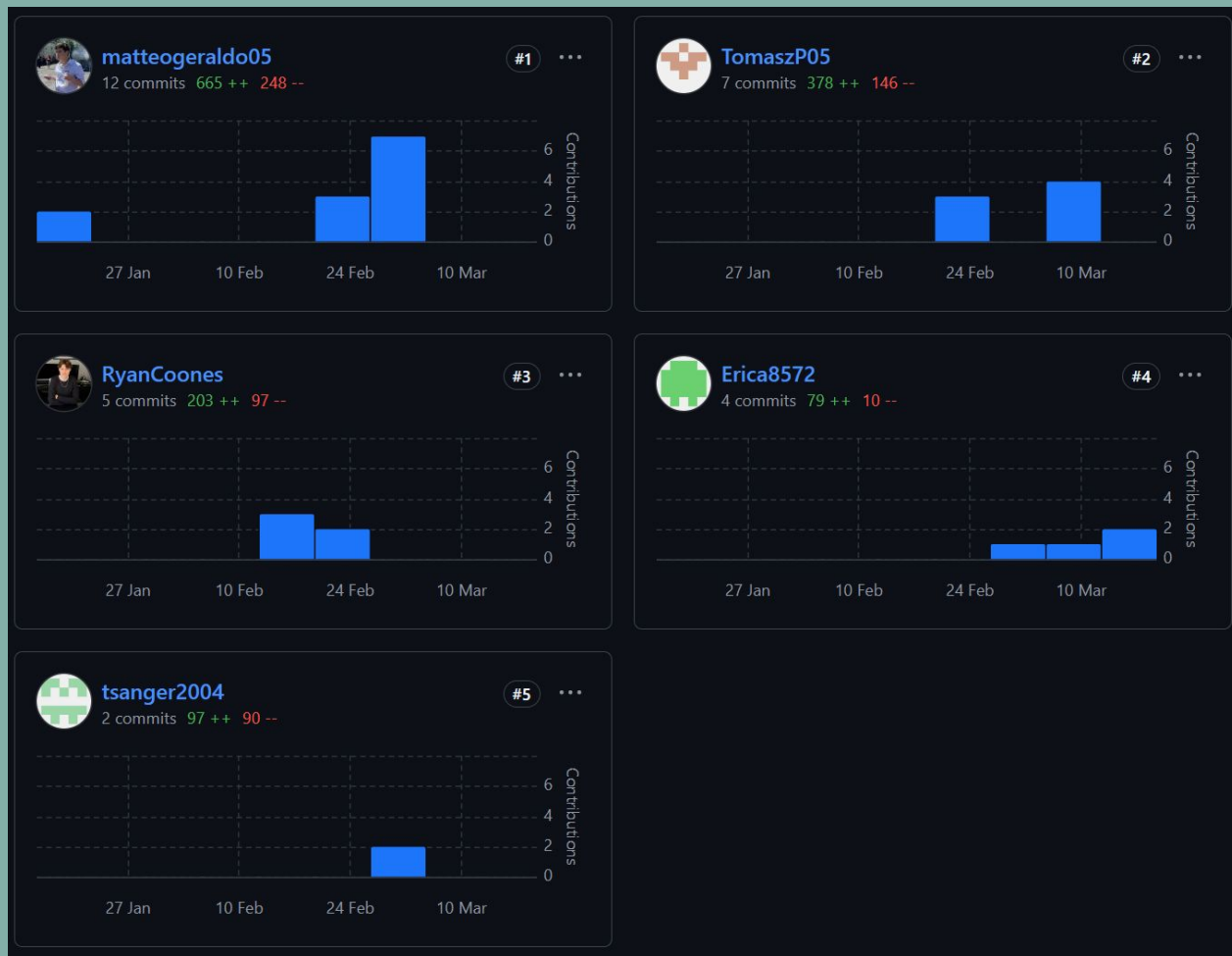
    A scrollable frame to house list of celebrities
    celebritiesFrame = CtkScrollableFrame(mainFrame,
    celebritiesFrame.pack(fill="both")

    Load celebrities from CSV file
    loadCelebrities = functions.load_celebrities_file("celebrities.csv")
```

REPO COMMITS



REPO COMMITTS



TERNOPS

Branches

Branches

New branch

Overview | Yours | Active | Stale | All

Q Search branches...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	2 hours ago			Default	...

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
unitTests	3 hours ago		2	3	...

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
gui	36 minutes ago		9	11	...
unitTests	3 hours ago		2	3	...
filtering	last week		2	1	...

5

Retrospective

```
if ! command -v python & \
then
    echo "Python3 is required"
    exit 1
fi

# Step 2: Check if pip is installed
if ! command -v pip &>/dev/null & \
then
    echo "pip is required to install dependencies"
    exit 1
fi

# Step 3: Set up the virtual environment
echo "Setting up virtual environment"
python -m venv venv

# Step 4: Activate the virtual environment
echo "Activating virtual environment"
source venv/Scripts/activate

# Step 5: Install dependencies
echo "Installing dependencies"
pip install -r requirements.txt

# Step 6: Run the GUI application
echo "Running the application"
python src/gui.py

# Step 7: Deactivate the virtual environment
echo "Deactivating virtual environment"
deactivate
```


ITERATION 2 RETROSPECTIVE

Went Well

- Better communication
- Ability to individually complete tasks during a busy time

Improvements

- More brief meetings
- More frequent pushing + pull requests to keep everyone on the same page

Challenges

- Writing tests
- Time management delays leading to crunch near the end

Resolution

- Ask clarification from team members
- Learn more about pytest

6 Questions and Answers

```
to_csv(file_path, index=False)

# Check if the path is a file or directory
if os.path.exists(images_path):
    if os.path.isdir(images_path):
        # If it's a directory, remove it
        try:
            shutil.rmtree(images_path)
            print(f"Successfully deleted folder: {images_path}")
        except Exception as e:
            print(f"Error deleting folder: {e}")
    elif os.path.isfile(images_path):
        # If it's a file, remove it
        try:
            os.remove(images_path)
            print(f"Successfully deleted file: {images_path}")
        except Exception as e:
            print(f"Error deleting file: {e}")

# Print a success message
print(f"Successfully deleted {images_path}")

# Define some functions to ease the process
def copy_image_to_folder(image_path, folder_path):
    """Copy an image to a folder"""
    Create new folder if it
```

```
if userAccount.get_user_name() == "Guest" :  
    # Login / Register buttons  
    loginButton = CTkButton(topFrame, text="Login", command=lambda: showSignInFrame("Login to Account"), width=60)  
    loginButton.pack(side="right", padx=15)  
    registerButton = CTkButton(topFrame, text="Register", command=lambda: showSignInFrame("Register Account"), width=60)  
    registerButton.pack(side="right")  
else:
```

MATTEO DEANGELIS GERALDO
RYAN COONES
TOMASZ PUZIO
ERICA PATEL
NOLAN TSANG

THANK YOU

TERNOPS