Title:
Can’t Forget Your Face

Who:
Greg James
Matt Ceriello
Michael Yon
Tyler Curnow

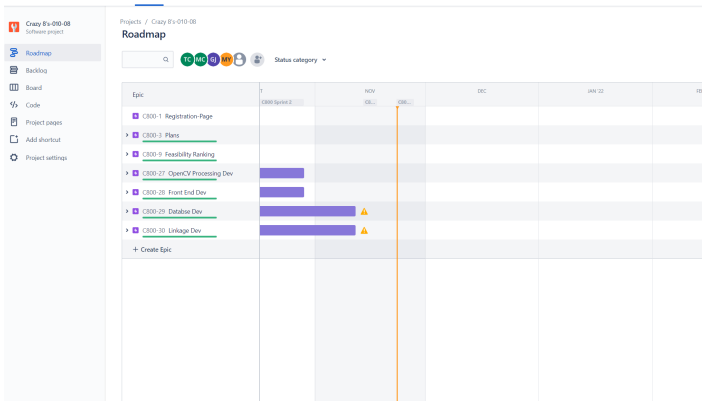
Project Description:

Can’t Forget Your Face is a website that when visited will prompt the user to enter a username and will train CV software to recognize their face. After training, the program recognizes that person's face, and will allow access to a blank text document where the user can write whatever they want. The facial recognition data, the username, and the document are persisted, so a user can log into their document at any time with their face. The practical application for this program is to allow users to keep their information in a safe and a secure spot where they can access it whenever they would like.

We were inspired to make this application because we found that regular Two-Factor Authentication (2FA) was too slow. When you would enable 2FA on a website you could often wait up to an hour to receive a text message or email confirming your account. Our solution is to allow a user to quickly and easily log into their account using their face, which you can never forget!

We decided to use this feature to let a user access a secure text document where they could store any information desired. However, that does not stop us from applying our solution to 2FA in other places.

Project Tracker:
Jira Board:
https://csci-3308-fall21-010-08.atlassian.net/jira/software/projects/C800/boards/1/backlog



Video:
https://www.youtube.com/watch?v=KBbrVDPI-4Y&t=2s

Git Repository:
https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-011-08

Contributions:

- Greg's contributions:

```
971a7fe (HEAD -> main, origin/main, origin/HEAD) done
48cb0fe no files
b3e35d9 file work
e68f008 Merge branch 'main' of https://github.com/CU-CSCI-3308-Fall-2021/CSCI-33
08-Fall21-011-08
381ba1f prettttty
02e6328 prettier
f6d84d5 push it
f2d1c00 login and textedit working
8776209 front end works
74a77dd Merge branch 'main' of https://github.com/CU-CSCI-3308-Fall-2021/CSCI-33
08-Fall21-011-08
3f68602 create account
bb22e36 pretty
1664163 pretty
e16bd2a textEdit
6b991ae starter code
2ee9efe fixed minorbug
0ff64d3 HREF work
26066d4 website files
8926eaf The folders shopuld exist now
8b91a8d Milestone 1
```

- Summary:

  Greg mainly worked on the front end at the beginning, starting with the basic layout then adding some CSS working with Tyler to make everything look pretty. He also created the wireframes for the original design. He then shifted over and created some database models and looked into starting a Postgres Database. Then, he worked with Matt and Michael to get all flask endpoints working and focused on the login flow to make the website more usable. He also worked on the sqlalchemy integration, querying for user info, and some work on saving user images.

- Michaels contributions:

```
21c5315 finished?
449c5fb Can create account only if face detected
7c30f4b merged with greg
cb9396e static pages working mc resolved
4c2a79f Pages are up and running
4933bea WIP connecting postrges and flask
56f2808 Flask serves data
c8eb996 Merge branch 'main' of https://github.com/CU-CSCI-3308-Fall-2021/CSCI-33
08-Fall21-011-08 into main
067ec7e Flask container working
```

- Summary:

  Michael mainly worked on planning out the architecture and getting Docker containers running properly to support the application while being developed. Early on he worked on developing a rudimentary browser based camera app and getting basic Docker containers working, and then moved on to building a custom image to support Flask, OpenCV, and SQLAlchemy. As the project came to a head, Michael worked with Greg

and Matt on the Flask endpoints, SQLAlchemy integration, websocket, and various other odds and ends. Throughout the project, Michael helped resolve merge conflicts between branches, and helped integrate proof of concept implementations into the Docker environment.

- Matt's contributions:

```
f91e072 Delete .matt1 directory
904ab1f Delete .morganfalter directory
f0b3935 Delete .llawson26 directory
e3ddb87 fixed login button
292d15d added FAQ page to nav bar
e5c0c8d button to verify face
b89f36d Python opencv linked w/ JS opencv for register
cda79bf navigating pages in flask
a6f276d database connected with flask-migrate
61e0620 Added facial regocnition to repo
8eb01d6 Updated Milestone 2 PDF
3af9d43 Added Milestone 2 PDF
c70523a Updated Readme (Michael's work)
```

- Summary:

  Matt mainly worked on implementing facial recognition functionality to the app using OpenCV. Early on, Matt worked on getting a working OpenCV program on his own machine. He then integrated that to the app once the front-end was up and running. Once the facial recognition was in a good spot, Matt helped Michael and Greg with Flask, along with getting the database set up using SQLAlchemy. Also, Matt implemented SocketIO functionality that would allow for camera access from the browser, instead of from wherever the program was being hosted.

- Tyler's contributions:

```
323c42c Finalized pages and fixed logo
5d318c2 Finalized page
88e8f8a added to wrong directory
585e519 added more text boxes and fixed delete account
21d1a32 added a few more text boxes and fixed delete acc
210573a Fixed minor errors and added a "current username"
7e00a08 Interlining and logo for all pages
0629e24 "Made the website look pretty" Also added logo
aa2f9f3 Add files via upload
45104e3 Cant Forget Your Face Logo
f198402 fixed account settings
661fa8c Created account settings HTML
9207fe3 FAQ page outline done
67dbe84 Add files via upload
```

- Summary:

  Tyler mainly worked on the frontend throughout the majority of the project. He helped Greg with setting up the wireframes and then moved on to start building each page in HTML. After each page was roughly layed out Tyler then worked to make sure that everything looked "pretty" and professional. With Greg's color scheme Tyler was also able to make a logo that was a perfect fit for his group's application. Once the front

end was close to being finalized Tyler worked with Greg to help find ways to integrate the front end with the backend endpoints. Finally he ran user tests to make sure that the front end of the website had a good flow.

Deployment:

Our app was deployed using docker. When first pulling the repository one must first do a docker-compose build command and wait for that to finish. It should take about a minute. After that simply run docker-compose up and the application will be live. It can be accessed on localhost:5000 through Chrome.