

Background to Project

At the beginning of my Level 3 first year, we had an induction which included a virtual tour. The tour showed us exciting works, images from other departments, but none from ours. This raised questions, curiosity and a desire to have images from our department on that tour.

Purpose

A website created by Level 3 IT students of Peterborough College (Grad. 2022), to collect student votes for the **Best Teacher** award to an IT lecturer.

Teacher Result Category

Because the department has different levels i.e Level 1, 2, 3. Some teachers only cover some level(s), teachers who are entered on the system would belong in either of two groups (not both);

- Level 1 and 2 - **Unique**
- Level 3 - **Higher**

Because of the split in level, there have to be two results for best teacher i.e **Best Unique Teacher** **Unique**, **Best Higher Teacher**.

However, students from certain levels cannot vote for a teacher from a different level i.e a level 1 or 2 students cannot vote for a *Higher* teacher. Therefore, the system has to be able to recognise the level a student belongs to. Once recognised, it should give them a unique code. This code would be used to determine the teachers they get access to vote for, an example of how the system would implement this is illustrated below;

Student Level	Code Type
Student selected level = 1 or 2	Code = U-1234567
Student selected level = 3	Code = H-1234567

After a student gets their unique code, they can proceed to vote, if their code starts with a **U**, then they would be presented with Level 1 and 2 teachers, if their code starts with an **H**, they would be presented with Level 3 teachers.

Decision Algorithm (DA)

To ensure the results are not vague or based on teacher popularity/favouritism and that it takes into account the complexity of the unit(s) covered by the teacher i.e the GLH and how the student feels about the difficulty of the unit(s) and other quality-related information. The system has to be more robust in judging how a teacher achieves the status of the **best teacher**. The processes of DA are outlined below;

- ☐ GLH for every unit a teacher teaches
 - ☐ Total GLH would be divided by 60 to give the final GLH score i.e 120 would be 2 points
- ☐ Unit difficulty score from student
 - ☐ Easy 1 point, Medium 2 points, Hard 3 points
- ☐ Does the teacher make the unit easy
 - ☐ Yes [student unit difficulty *2], No

- ☐ How supportive is the teacher
 - ☐ Weak 1 point, Solid 2 points, Devine 3 points
- ☐ Their teaching style
 - ☐ Good 1 point, Average 2 points, Devine 3 points
- ☐ Does the teacher provide useful resources that help with your learning
 - ☐ Good 1 point, Average 2 points, Devine 3 points
- ☐ Does the teacher pay attention to every student
 - ☐ Weak 1 point, Solid 2 points, Devine 3 points
- ☐ How strict is the teacher?
 - ☐ Fair, Too harsh, Does not care
- ☐ Organising group works and practical
- ☐ Does not make the student do too much work
- ☐ Teacher getting mad for no reason

Goal

To create a working website where students can vote for the best teacher.

Considerations

- Student votes must be accurately recorded.
- Votes must be anonymous.
- A student can only vote once.
- The home page could have a countdown to the result day.
- Requires a domain.
- Use the college logo and colour.

Plan

To achieve the goal we are relying on contribution and time investment from students, resource help from teachers.

The project would be developed and deployed using *online IDE Replit* and **GitHub**. Both platforms provide remote access, team collaboration, version control, testing capabilities, web hosting and more.

Website to have five pages; home, code, voting, success vote and results page. See [HTML Tasks](#), [Wireframe](#).

Requirements

1. Unique code generated for students.
2. Student access voting website.
3. Unique code verified.
4. The student chooses a teacher to vote for.
5. The student receives confirmation of the recorded vote.
6. Anyone can check the results.

[Wireframe / Requirement video \(42 sec\)](#)

Initiators / Contributors

- Ighomena Daniel Odebala (IDO)
- Conor O'Leary (CO)
- Adam Young (AY)
- Matt Clark (MC)
- Daniel Estima (DE)
- Alisha Chaudhry (AC)
- Ethan Walker-Smith (EWS)

Jobs

- HTML
- CSS
- JS
- Wireframe/Graphic Design

Project Aspirations

Aspiration	Possibility
Project to be maintained annually	Low - Medium
Project could be proposed to other departments of the college	Low

Why Contribute

- Work within a team, share **IDEAS**, **LEARN** and **GROW** together.
- A chance to boost **CONFIDENCE** for some and create an edge when applying to universities or career progression.
- A chance to showcase talents, develop and improve **IN-DEMAND SKILLS**.
- Gain experience in developing and managing a **LIVE** website.
- Polish your **CV** with your achievement in your role.
- Make your **UCAS** statement robust.
- Come have **FUN** with us.

HTML Tasks

Task	Subtasks	Currently done by	Status
Create "Home" page	• Add "Cast A Vote" button • Add "See Results" button	IDO	completed
Create "Code" page	• Add "Please enter your code to vote" as label or paragraph • Add Input field with "Enter Unique Code"	CO	completed

	- as placeholder - Add "Cast Vote" button		
Create "User Voting" page	- Add "Please choose who you are voting for" as label or paragraph - Add Four (4) radio buttons with data; "John Doe, Nancy Doe, Bridget Doe and Nathan Doe" - Add submit button for radio buttons with "VOTE" as value	CO	ongoing
Create "User Success Vote" page	- Add "Success" division i.e div - Add "You have successfully voted" as paragraph - Add "See Results" button		
Create "Results" page	- Add Four (4) paragraphs with data; "John Doe, Nancy Doe, Bridget Doe and Nathan Doe" - Add "Home" button - Add "Vote" button		

CSS Tasks *

- Position page contents to meet design
- Add styling to pages (colours, padding, border, margin)
- Style buttons

JS Tasks (Possibly PHP) *

- Generate unique code for students
- Check and validate code entered by student
- Record student votes
- Compare votes to determine winner

Links

- [Wireframe](#)
- [Temporary Test Link](#)
- [GitHub Repo](#)
- [Replit Link](#)

Testing

**Archie Bradford**Tue, 7 Jun, 18:54 (7 days ago)☆ ↶ ⋮

to me ▾

Hi Ighomena,

After testing out the website I must admit that a lot of hard work has gone into that and it pays off because you can see how amazing it is! Really well designed and works well.

Thank you for sending it to me for me to try.
Hope this helps

Many thanks
Archie

**Callum Riches**Tue, 7 Jun, 18:58 (7 days ago)☆ ↶ ⋮

to me ▾

I think you have done an excellent job on the website all the buttons work correctly and clearly show where you need to go for each section.

As there isn't much to be able to add to the website I think the best step forward is to add more CSS colours and possible rearrange the position and size of the buttons.

Another idea I have is making the final results locked and only visible to an admin or only you so you can see the final results once everyone has voted.
Also if possible add images of the teachers (if allowed), or other relevant photos like computers or other IT devices.
Another potential idea is having a percentage graph to show the percentage of people that voted for each teacher.

* Other testers gave verbal feedback

Evaluation

Start date: 19 April 2022

End date: 13 June 2022

Cost: £0

Contributors: 7

Testers: 7

Initially the project started with a static website approach and it had contributions from various students as jobs were split and completed on time, however, as the project progressed it grew past being a static website because it needed to store and read data, compare data, run processes and send emails. With the new form the project had progressed to, we had to implement a system capable of running all those functions.

Initially, I considered using PHP, but I had no experience with PHP and felt it would be too complex for me and would require more time, I finally decided to use Django (A python web framework), which I have some experience with and very comfortable with Python. This made the task easy and not too complex in theory.

I used Django in developing the website and used Heroku to host it, I also used Github for version control and backup. I had no major issues during development.

The project was launched, although the launch was delayed due to college work (assignments) that needed to be completed. After launch, we recorded votes from 30 students, and during the analysis of the data, it was found that these students were level 3.

Lessons learned

Linking custom domain

I managed to secure a free custom domain (ieguestudentvote.tech, this was acquired through Github's Student Developer Pack which gives students access to freebies such as some 1-year free domain and access to other world-class technology packages.

After getting the domain, I successfully linked the websites, but it had no SSL and the domain was insecure because I was using a free cloud hosting service (**Heroku**) and I could only get an SSL certificate using a paid subscription. The next option for me was to use **Cloudflare's** SSL.

I had some experience with using Heroku and Cloudflare which was successful for me in the past for a **Flask** application, but unfortunately, it did not work this time with a **Django** application. And because I did not allocate enough time to sort the domain, I could not make the domain secure for usage during or after launch.

Using environmental variables

Generally, website security is taken seriously, mistakes I made initialising the project was pushing some sensitive data to Github, I later learned that these data should be stored as an environment variable on the development machine or hosting service by storing those data to the local space and calling a reference to them when needed. Before launching the project I implemented environmental variables and replaced the initial exposed data.

In the future, it would be best to set up environmental variables as soon as development starts, as the overall security of the website would be improved and time would be saved on going back to replace exposed data.