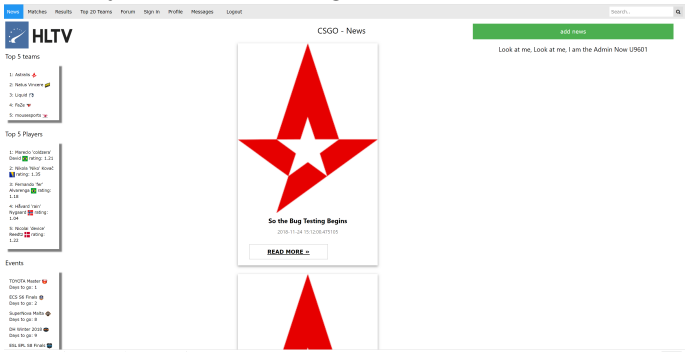
Advanced Web Tech Report

Christopher Jones
40274924@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

HLTV - CSGO Coverage Clone

1 Introduction

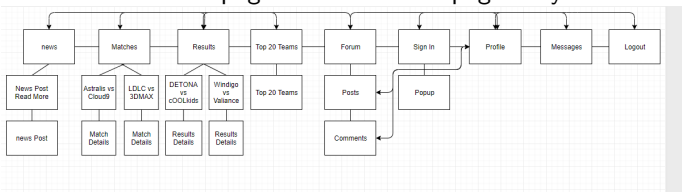
I based my web app on the very popular Counter-Strike Global Offensive coverage news site, which is I get all the news about whats going on in the CS community, matches being played and results that came from those matches. The main page of my web app is where all the news articles are saved and can be read if the user clicks the read more button. All of these posts are written by the admin which is of course me. The home page also has a nav bar at the top so the user can navigate their way around the website, more options become available to the user once they have registered and logged in allowing them to access the forum and their profile of which they can edit. This also allows them to write a post on the forum page and then they can right a comment about that post along with others. The user can also choose to like a post if they feel it is a good post or if they realize it wasn't as good as they thought then they can also unlike it.
They can also edit their profile to have an about me section so they can have a little bio and if they don't like their user name they can chose to change it and such.



As you can see above the homepage includes the news in the center column with them going newest to oldest down the middle. On the right you can see a green button which only admins can see and it is how I create the news articles for the users. On the left column you see there are 3 boxes, the top shows the top 5 teams in the world currently in their ranking order. The next box shows the top 5 rated players of 2017 because 2018's aren't out yet. It doesn't work by ratings even though you could have a higher rating doesn't mean you will come higher than someone else with a lower rating.
The final box is shows what events are coming up in the next couple weeks or so allowing the user to see what they got to look forward to. The web app also allows the user to see the results of games but then embed the Twitch/YouTube URL so the user can watch the game being played live or watching the VOD of a game that has already been played. It also breaks down each map to show the individual score of both teams on their CT and T sides on the maps they played (includes Over Times as well).

2 Design

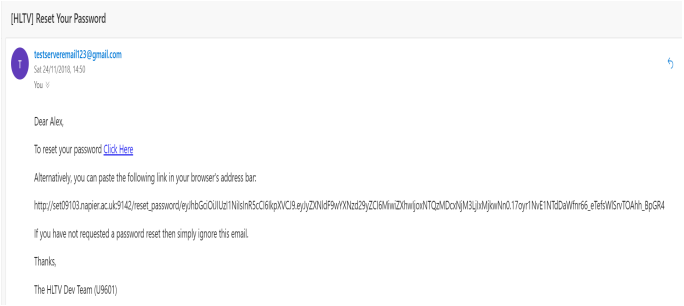
I used a web design hierarchy for my web page as that is the best for the user to navigate though my website and it also makes it easier for them to return back to the home page or to another page they were at.



As you can see from the diagram above every web page has its own nav bar which means every page is linked. However, if you aren't logged in then you wont be able to see Sign in because those pages require a user to have an account. This is to stop people from breaking things like the forum and stuff. The structure of the top 20 teams is structured in such a way that it allows the user to nicely see what teams are the best in the world. only included 2 match details and 2 results details to make the diagram readable because there is like 50 combined or there about. So under those sections there would of course be more boxes and more lines going to them but they only connect back to the nav bar and not to any other matches or results

As you can see from the above image the teams are in a table that has been made to have no borders. So this is so the data is stored in a nice and neat fashion on the page. It also shows the user who the teams are, there players

and how many points they have acquired from playing in tournaments. I have made it so that there is no login page as such on my web app there is just a pop up that will appear on the page you're currently on if you click sign in on any of the web pages. It will also give you an option to register if you're not currently a user which does take you to another web page.
The users password is hashed using werkzeug security when they register their account so no one can see that users password in the database. This decreases the chances of someone hacking your account on my website. If you try and click on forum but you aren't currently logged in then it will take you to a login page because mainly I couldn't find a way of making the JavaScript run to open the pop up but the pop up is the main way to log in. If the user has made a user but, has unfortunately forgot their password then there is an email reset function in the website that allows the user to type in their email they used to create the account and then it will send them an email reset email to that email.

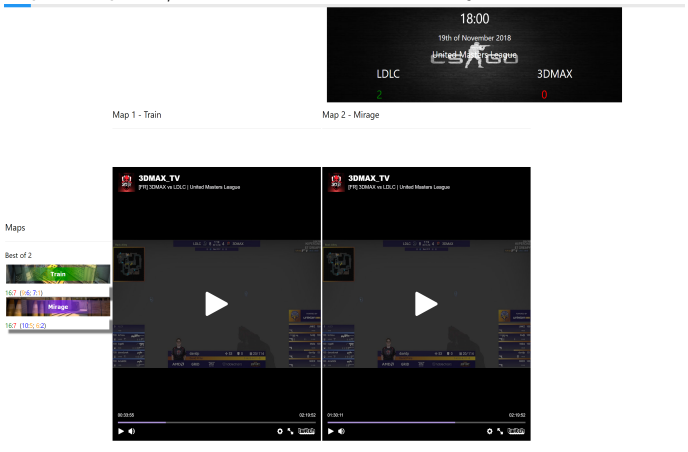


The image above shows the email a user gets once they have requested a reset password email then the user has to clicked on the email reset url. Depending on which email service you have will depend which link you will click on things like gmail seem to work with both whereas outlook only likes the top one. Once clicked they will be instructed to change their password and then re log back into the website with their new password they have chosen.
Every web page bar the the match details and result details all have the page split into 3 columns, this is so I can have the HLTV logo on the left, all the actual import data in the middle and the admin related stuff like the add news button on the right. This column structure is very prominent on the pages that have posts on them as they are all centered in the middle.

As you can see from the above photo the posts are centered in the middle of the page. This does leave a lot of white space but that photo was taken on a 4k display of which I didn't do the coursework on. It does mean that the data is nicely centered the page and easy to read and understand because it isn't all over the place. As I said the only 2 web pages that don't have this structure on is the match details and the results details this is because the columns weren't big enough so had a lot of collision issues mainly because the twitch windows are quite big.
So instead of this structure I had to use border less tables again as that was the only way to order the data in a some sort of nice way. The Posts section also give the user the ability to delete and edit posts that they may have messed or don't want other people to see. This is also the same for the comments as well along with being able to like the post. Of course so people don't delete each others comments and posts you can only delete your own posts and comments Others users can also follow other users and the website will tell you how many people are following you and how many people you follow. The final bit of the design is the edit profile button which is located on your profile and can only be edited my the user with that user name. It allows the user to have an about me section like a biography about them and shows the last time they logged into the website itself.
I used a tutorial by Miguel Grinberg Tutorial which is a huge Flask tutorial of which I was able to get some of the features off and use in my web app as his tutorials are very, very well documented and helpful when it came to how to use the database with my web site. This meant I could had a reference to go to if something wasn't working or I needed to know how to do something because everything you could ever need to know about python flask is in that tutorial. It also meant I could focus my efforts on other parts of the design rather than getting bogged down the database and the server side python stuff.

3 Enhancements

I think one of the main features I would like to change is the design and feel of the web app by using bootstrap. I would have liked to have used with this and the previous coursework but felt as though as it is a tad cheaty and tried to create my own css which wasn't actually too bad but seeing what you can do with bootstrap has changed my mind. I think I would have liked to have enhanced the match details, result details and perhaps the posts/comments but I think they are fine for now.



As you can see from the above picture is one of the results

page which I think could have used a bit more styling in some places especially with the banner at the top which I couldn't find a good image for so just left it as that and didn't end up changing it in the end anyway.
A feature I really would have liked to have added but couldn't get working was the news posts having their own image rather all having the astralis logo. I managed to get to work if it were to be a profile picture but didn't know how to uniquely identify the photos for the news so after much trail and error and breaking of the migrations in sqlite3, decided to give up and just focus my efforts else where. would have also liked to improve on how much white space there is on some of the web pages as although the columns worked they didn't allow for a lot of customization because of their boundaries. On pages that only have like the register there isn't much going on and its a bit bland and boring. Another feature I would have liked would have been to the ability to like comments as well as profiles but due to time constraints I didn't have the time to do this but would have been easy to implement as I already had the foundry for it with the comments on posts. would have really liked to fix the Direct messages post section as it really is all over the place but it didn't let me edit it so couldn't change it because it was half bootstrapped as I needed a form() from it.
I think one thing that heavily contributed to time constraints was the fact that I didn't have a base.html file and copied and pasted the nav bar about 10 times and every time I had a change I had to re copy and re paste it which took up so much time. This could have been easily sorted my simply using an extends jinja tag. The last thing I think I would have liked to have enhanced, was I did have an API that used beautifulsoup to read the divs from HLTV itself however after several of days of working with this API it decided to stop working for some reason as the divs in htlv.org never changed

4 Critical Evaluation

I feel the best looking part of the website in terms of styling has to go the Top 20 teams as I think that worked so well. It just its easy on the eyes and shows all the data that I wanted it show in a nice format. An area that really didn't work well was the result details as I said before it just doesn't look good what so ever but I really went for functionality over design as that is what I'm good at.
Another thing that worked poorly had to be the API which I had tried and tested before starting this coursework but as I started getting the data from it decided it no longer wanted to work which left me in a bit of a pickle with how I was going to get the data for the matches and results. So to actually have matches and results I wrote them into a JSON file instead which means they wont update on the fly like they would with the API but at least there is something to display. One thing that I think actually did work well was the profile pages and the comments even though they have a ton of white space they aren't actually half bad for how basic they are.

As you can see from the above image is the profile of myself because I am the admin. It shows your profile in the left column and all your posts in the centre column of which users can go and like or comment on from the profile page.
I think this has worked out quite well as although its basic it has quite a lot of stuff going on it which makes it look semi good. Another thing that worked well would probably be the news page even though there is some spacing problems it has all the data that is meant to have bar the betting like the real HLTV has but, actually has everything they have if not just in a more simple manner. Again, if the API had worked then the boxes like the top 5 teams and such would update automatically as soon as the HLTV ones updated but this decided not to be the case in the end which sucks.
Another bit of functionality that I would liked to have got to work would have been the search bar which I really just ran out of time for but may look into doing after this coureswork is done a dusted as it would be a good thing to try and implement.

5 Personal Evaluation

I think my performance for this coursework was actually pretty good as I put a lot of effort in to hopefully get a decent mark. The only thing I think I would have liked to have worked more would have been some of the design as I have said above just mainly to make the website look at tad better in some places. I did put a lot of hours into this project even when I went on holiday I did quite a substantial amount of work before I went to bed and what not.
I think the main challenge I had to overcome was the API issue because that was what my website was based around as I was going to save the data to the database so I could have a lot of games built up on the web page but that unfortunately could not be accomplished. I overcame this by writing JSON files the matches on certain days and just displaying them instead even if it is very inefficient it was the only way to do other than manually put them into the database which would have caused other problems.
Talking of databases that brings me to my next point which is where I managed to do a migrate on the database perfectly fine but then forgot to do an upgrade on it which meant it was just holding onto that migration. This meant if I were to run the migrations command again there would be 2 different migrations and the database wouldn't know what to do and gave you an saying "The database is out of sync with the

current migration". To solve this problem I had to delete the migrations folder and then delete app.db which meant I lost all the data in the database but could actually use the thing. This was honestly quite annoying bug, it also happened if you delete a column in the database for some reason but I think that was down to me having multiple terminals open at the same time and the database got cached on one of them some how.
. Another problem I had was trying to get my head around the way jinja could he used to get data out of say a list because at first I was unsure how the data could be passed from Flask into the templates but, after finding out how Jinja worked I was able to create a lot of cool things with it. This meant I could pass my JSON files across and such without much hassle.
Overall I think I performed pretty well considering the set backs I had at the beginning of the project but pulled through to create a project that, in my eyes is pretty good and I'm happy with it.

References

[1] Stack OverFlow for General Stuff,
`https://stackoverflow.com/`

[2] The python Flask work book on moodle,
`https://moodle.napier.ac.uk/course/view.php?id=28467`

[3] The MEGA Python flask tutorial by Miguel Grinbery,
`https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world`

[4] The Jinja work book on moodle,
`https://moodle.napier.ac.uk/course/view.php?id=28467`

[5] w3Schools for some HTML and CSS,
https://www.w3schools.com/

[6] Google Images for the images I downloaded,
https://https://www.google.co.uk/images

[7] Majority of photos were from hltv
https://hltv.org

[8] CSS Font Stack for the Ubuntu font that I never got working D:,
https://www.cssfontstack.com/Ubuntu/