Final Project Report

Subject: Web Technologies - II UE17CS353

Project Title: APP Gaming

Project Team: Pradyumna YM PES1201700986

Pruthvish E PES1201701629

Anush V Kini PES1201701646

Project Description

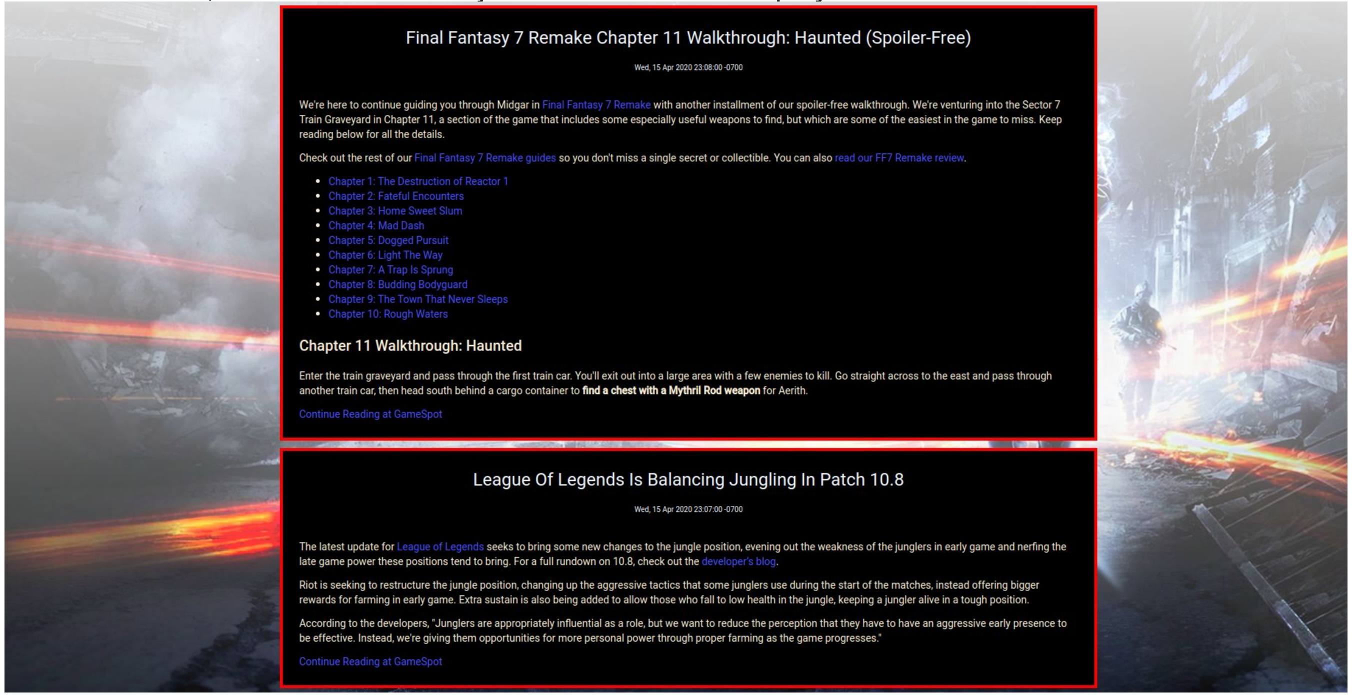
- We have implemented a one stop gaming website using Angular with the following features.
    - Play Games
    - Login System with authentication using JWT and bcryptJS
    - Friends System
    - Global Chat
    - Multiplayer Real Time Gaming (for Chess)
    - RSS feeds
    - Synergy scores (recommender system)
    - Smart profile ordering (using transfer learning CNNs)
    - REST API in backend using Nodejs
    - All of them developed from scratch by us.

Technologies Used

- Programming Languages
  - Python
  - Typescript/Javascript
- Tools
  - Angular
  - Node JS
  - Tensorflow 2.0
  - Bootstrap
  - Mongodb

Techniques Implemented

- **RSS feeds** for the blog section of our website. We obtain the RSS feed from a gaming website, parse it on the server, and serve JSON objects to the client to display the latest information.



Final Fantasy 7 Remake Chapter 11 Walkthrough: Haunted (Spoiler-Free)

Wed, 15 Apr 2020 23:08:00 -0700

We're here to continue guiding you through Midgar in Final Fantasy 7 Remake with another installment of our spoiler-free walkthrough. We're venturing into the Sector 7 Train Graveyard in Chapter 11, a section of the game that includes some especially useful weapons to find, but which are some of the easiest in the game to miss. Keep reading below for all the details.

Check out the rest of our Final Fantasy 7 Remake guides so you don't miss a single secret or collectible. You can also read our FF7 Remake review.

- Chapter 1: The Destruction of Reactor 1
- Chapter 2: Fateful Encounters
- Chapter 3: Home Sweet Slum
- Chapter 4: Mad Dash
- Chapter 5: Dogged Pursuit
- Chapter 6: Light The Way
- Chapter 7: A Trap Is Sprung
- Chapter 8: Budding Bodyguard
- Chapter 9: The Town That Never Sleeps
- Chapter 10: Rough Waters

Chapter 11 Walkthrough: Haunted

Enter the train graveyard and pass through the first train car. You'll exit out into a large area with a few enemies to kill. Go straight across to the east and pass through another train car, then head south behind a cargo container to **find a chest with a Mythril Rod weapon** for Aerith.

Continue Reading at GameSpot

League Of Legends Is Balancing Jungling In Patch 10.8

Wed, 15 Apr 2020 23:07:00 -0700

The latest update for League of Legends seeks to bring some new changes to the jungle position, evening out the weakness of the junglers in early game and nerfing the late game power these positions tend to bring. For a full rundown on 10.8, check out the developer's blog.

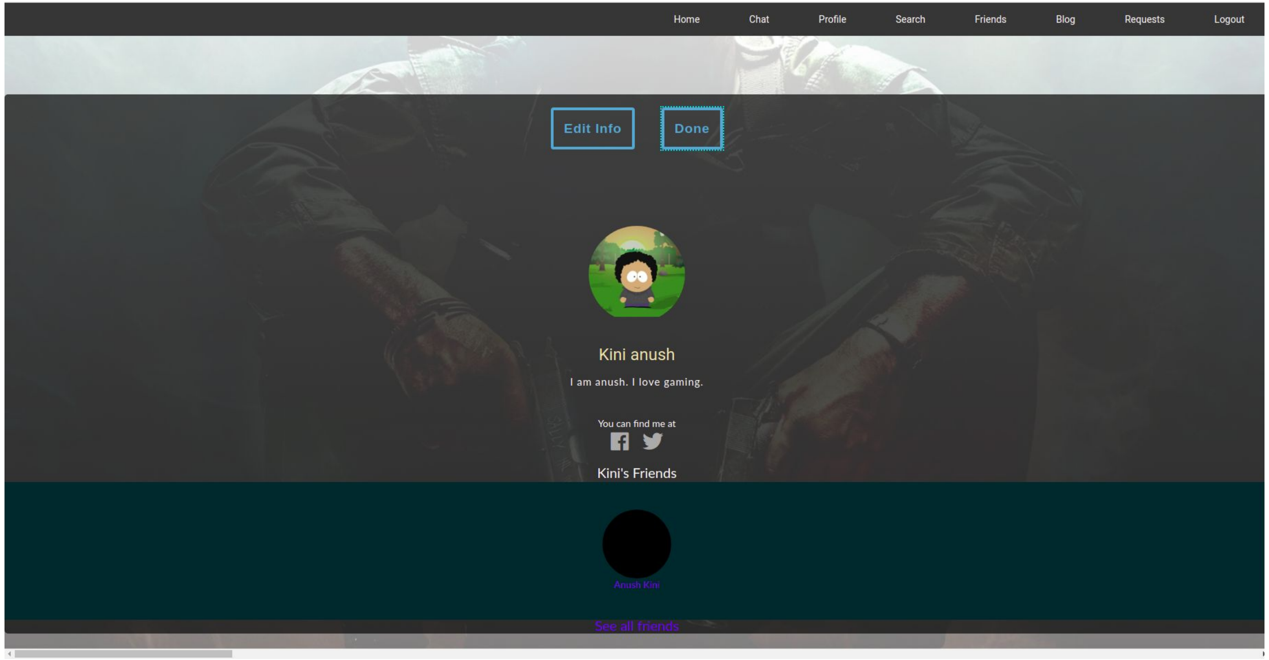
Riot is seeking to restructure the jungle position, changing up the aggressive tactics that some junglers use during the start of the matches, instead offering bigger rewards for farming in early game. Extra sustain is also being added to allow those who fall to low health in the jungle, keeping a jungler alive in a tough position.

According to the developers, "Junglers are appropriately influential as a role, but we want to reduce the perception that they have to have an aggressive early presence to be effective. Instead, we're giving them opportunities for more personal power through proper farming as the game progresses."

Continue Reading at GameSpot

Techniques Implemented

- **Multistage Download** : profile pictures route of the client uses multistage download, where the details of the user is retrieved from the database first, and then the profile picture of the user is retrieved, and in the next stage, his friend profile details are retrieved, and lastly, his friend's profile pictures are obtained.



Kini anush

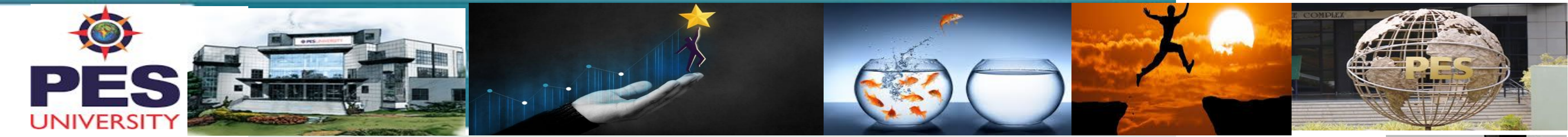
I am anush. I love gaming.

You can find me at

Kini's Friends

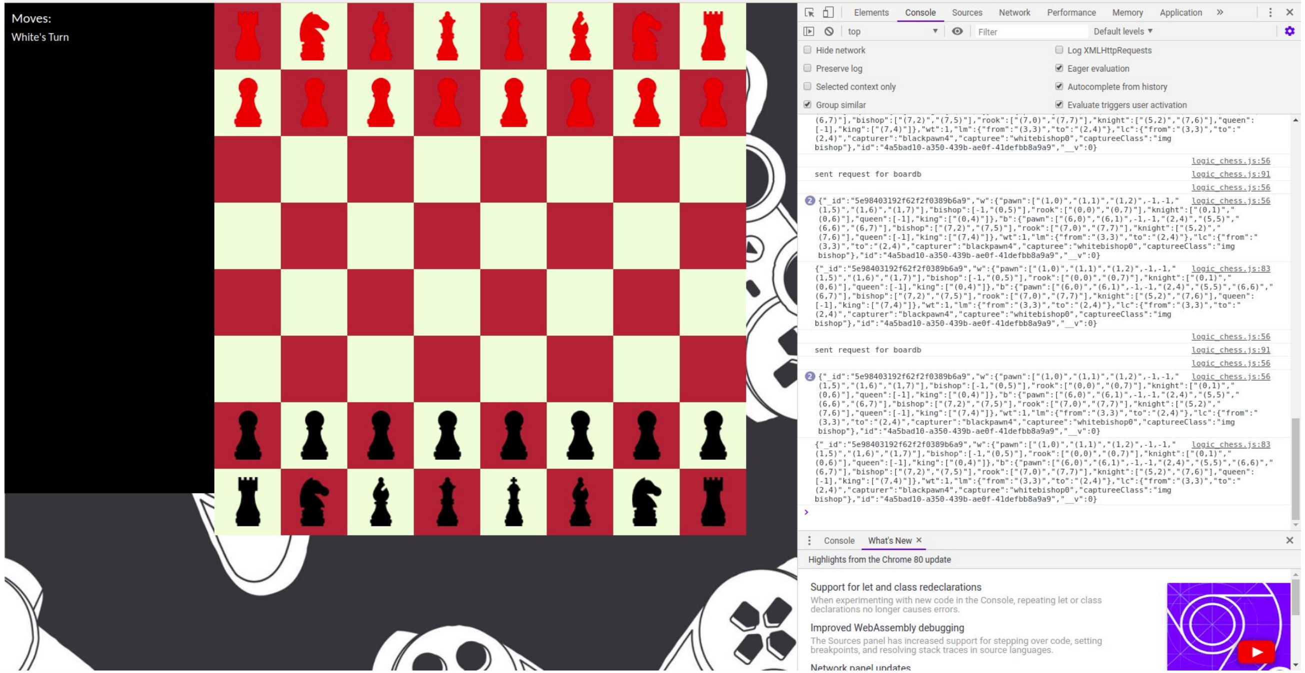
Anush Kini

See all friends



Techniques Implemented

- **Short polling** used for chess game : the chess client polls the server constantly, waiting for the other user to make his move, by making use of short polling AJAX requests every 500 milliseconds.



Moves:
White's Turn

- [ ] Hide network
- [ ] Preserve log
- [ ] Selected context only
- [x] Group similar
- [ ] Log XMLHttpRequests
- [x] Eager evaluation
- [x] Autocomplete from history
- [x] Evaluate triggers user activation

```
(6,7)"],"bishop":["(7,2)","(7,5)"],"rook":["(7,0)","(7,7)"],"knight":["(5,2)","(7,6)"],"queen":
[-1],"king":["(7,4)"]},"wt":1,"lm":{"from":"(3,3)","to":"(2,4)"},"lc":{"from":"(3,3)","to":"
(2,4)","capturer":"blackpawn4","capturee":"whitebishop0","captureeClass":"img
bishop"},"id":"4a5bad10-a350-439b-ae0f-41defbb8a9a9","__v":0}
                                                                          logic_chess.js:56
sent request for boardb                                                   logic_chess.js:91
                                                                          logic_chess.js:56
2 {"_id":"5e98403192f62f2f0389b6a9","w":{"pawn":["(1,0)","(1,1)","(1,2)",-1,-1,"   logic_chess.js:56
(1,5)","(1,6)","(1,7)"],"bishop":[-1,"(0,5)"],"rook":["(0,0)","(0,7)"],"knight":["(0,1)","
(0,6)"],"queen":[-1],"king":["(0,4)"]},"b":{"pawn":["(6,0)","(6,1)",-1,-1,"(2,4)","(5,5)","
(6,6)","(6,7)"],"bishop":["(7,2)","(7,5)"],"rook":["(7,0)","(7,7)"],"knight":["(5,2)","
(7,6)"],"queen":[-1],"king":["(7,4)"]},"wt":1,"lm":{"from":"(3,3)","to":"(2,4)"},"lc":{"from":"
(3,3)","to":"(2,4)","capturer":"blackpawn4","capturee":"whitebishop0","captureeClass":"img
bishop"},"id":"4a5bad10-a350-439b-ae0f-41defbb8a9a9","__v":0}
{"_id":"5e98403192f62f2f0389b6a9","w":{"pawn":["(1,0)","(1,1)","(1,2)",-1,-1,"   logic_chess.js:83
(1,5)","(1,6)","(1,7)"],"bishop":[-1,"(0,5)"],"rook":["(0,0)","(0,7)"],"knight":["(0,1)","
(0,6)"],"queen":[-1],"king":["(0,4)"]},"b":{"pawn":["(6,0)","(6,1)",-1,-1,"(2,4)","(5,5)","(6,6)","
(6,7)"],"bishop":["(7,2)","(7,5)"],"rook":["(7,0)","(7,7)"],"knight":["(5,2)","(7,6)"],"queen":
[-1],"king":["(7,4)"]},"wt":1,"lm":{"from":"(3,3)","to":"(2,4)"},"lc":{"from":"(3,3)","to":"
(2,4)","capturer":"blackpawn4","capturee":"whitebishop0","captureeClass":"img
bishop"},"id":"4a5bad10-a350-439b-ae0f-41defbb8a9a9","__v":0}
                                                                          logic_chess.js:56
sent request for boardb                                                   logic_chess.js:91
                                                                          logic_chess.js:56
2 {"_id":"5e98403192f62f2f0389b6a9","w":{"pawn":["(1,0)","(1,1)","(1,2)",-1,-1,"   logic_chess.js:56
(1,5)","(1,6)","(1,7)"],"bishop":[-1,"(0,5)"],"rook":["(0,0)","(0,7)"],"knight":["(0,1)","
(0,6)"],"queen":[-1],"king":["(0,4)"]},"b":{"pawn":["(6,0)","(6,1)",-1,-1,"(2,4)","(5,5)","
(6,6)","(6,7)"],"bishop":["(7,2)","(7,5)"],"rook":["(7,0)","(7,7)"],"knight":["(5,2)","
(7,6)"],"queen":[-1],"king":["(7,4)"]},"wt":1,"lm":{"from":"(3,3)","to":"(2,4)"},"lc":{"from":"
(3,3)","to":"(2,4)","capturer":"blackpawn4","capturee":"whitebishop0","captureeClass":"img
bishop"},"id":"4a5bad10-a350-439b-ae0f-41defbb8a9a9","__v":0}
{"_id":"5e98403192f62f2f0389b6a9","w":{"pawn":["(1,0)","(1,1)","(1,2)",-1,-1,"   logic_chess.js:83
(1,5)","(1,6)","(1,7)"],"bishop":[-1,"(0,5)"],"rook":["(0,0)","(0,7)"],"knight":["(0,1)","
(0,6)"],"queen":[-1],"king":["(0,4)"]},"b":{"pawn":["(6,0)","(6,1)",-1,-1,"(2,4)","(5,5)","(6,6)","
(6,7)"],"bishop":["(7,2)","(7,5)"],"rook":["(7,0)","(7,7)"],"knight":["(5,2)","(7,6)"],"queen":
[-1],"king":["(7,4)"]},"wt":1,"lm":{"from":"(3,3)","to":"(2,4)"},"lc":{"from":"(3,3)","to":"
(2,4)","capturer":"blackpawn4","capturee":"whitebishop0","captureeClass":"img
bishop"},"id":"4a5bad10-a350-439b-ae0f-41defbb8a9a9","__v":0}
>
```

Highlights from the Chrome 80 update

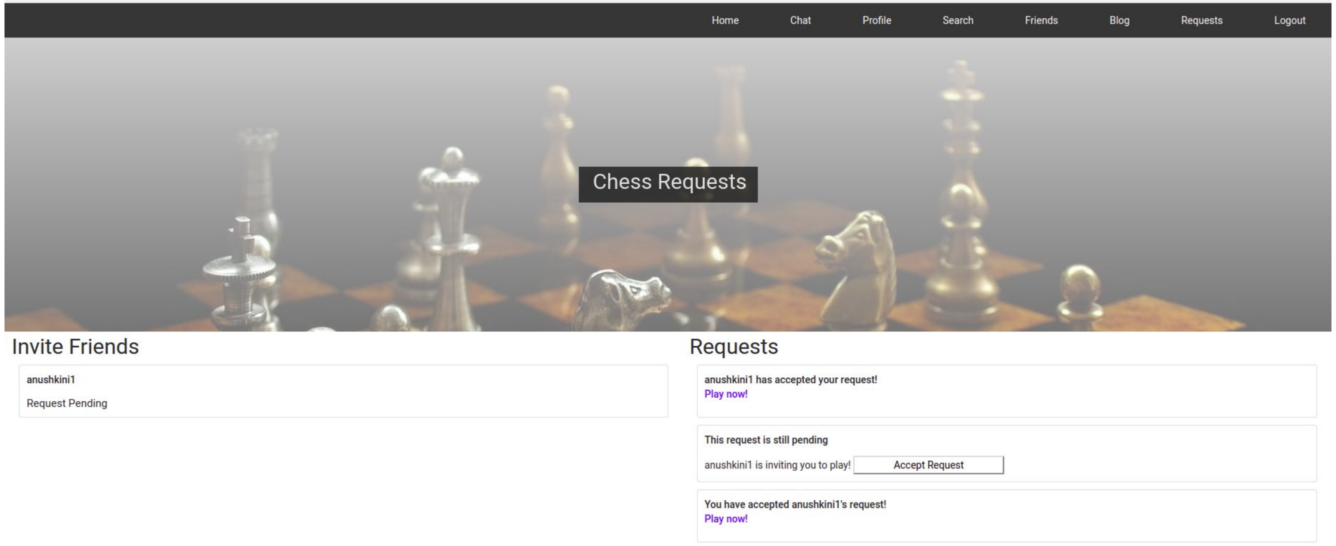
Support for let and class redeclarations
When experimenting with new code in the Console, repeating let or class declarations no longer causes errors.

Improved WebAssembly debugging
The Sources panel has increased support for stepping over code, setting breakpoints, and resolving stack traces in source languages.

Network panel updates

Techniques Implemented

- **Periodic refresh** : used in the chess requests component, where the status of the game requests get updated every 5 seconds.



Invite Friends

anushkini1

Request Pending

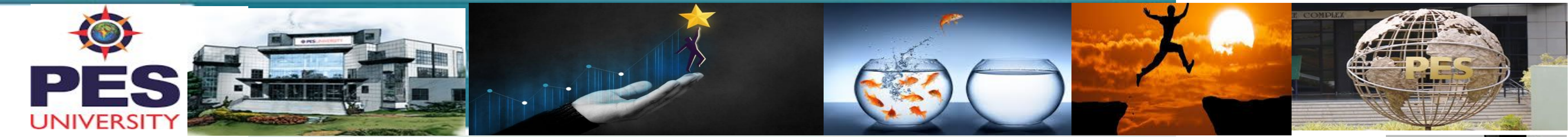
Requests

anushkini1 has accepted your request!
Play now!

This request is still pending

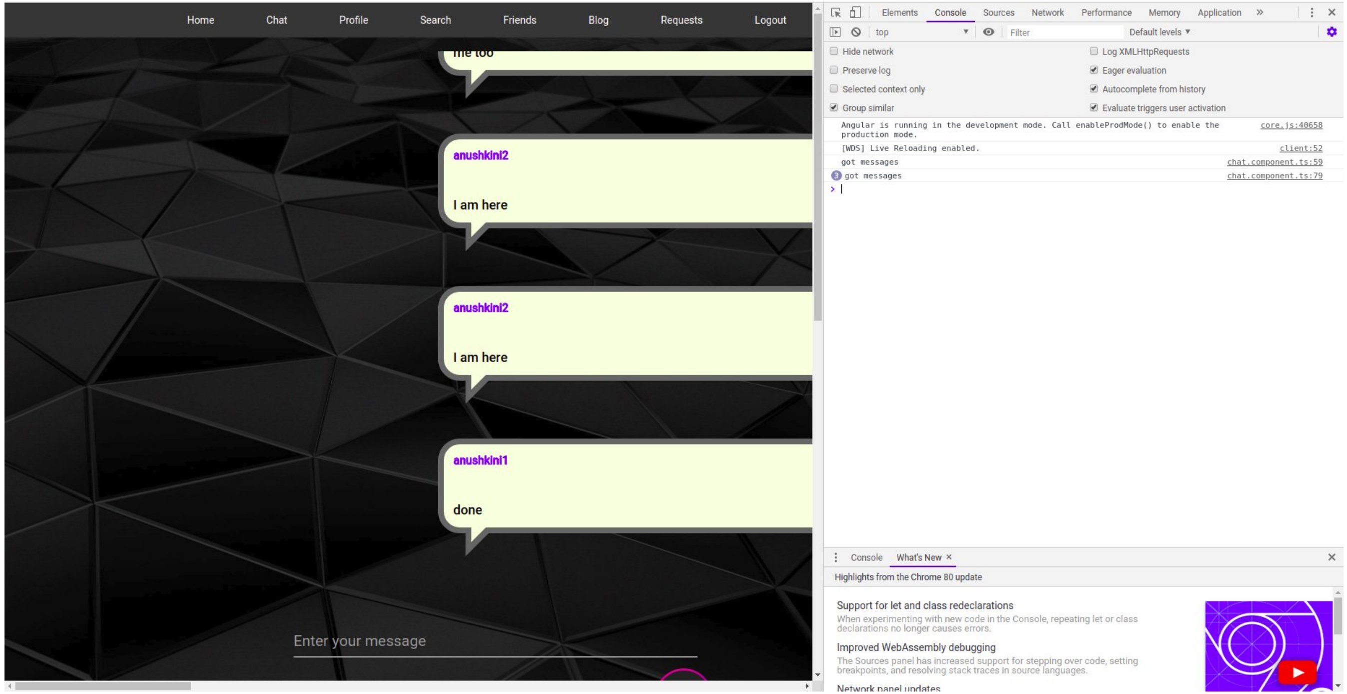
anushkini1 is inviting you to play! Accept Request

You have accepted anushkini1's request!
Play now!



Techniques Implemented

- **Periodic refresh** : also used in the chat section of the website.



me too

anushkini2

I am here

anushkini2

I am here

anushkini1

done

Enter your message

- [ ] Hide network
- [ ] Preserve log
- [ ] Selected context only
- [x] Group similar
- [ ] Log XMLHttpRequests
- [x] Eager evaluation
- [x] Autocomplete from history
- [x] Evaluate triggers user activation

```
Angular is running in the development mode. Call enableProdMode() to enable the     core.js:40658
production mode.
[WDS] Live Reloading enabled.                                                          client:52
got messages                                                                 chat.component.ts:59
3 got messages                                                               chat.component.ts:79
>
```

Highlights from the Chrome 80 update

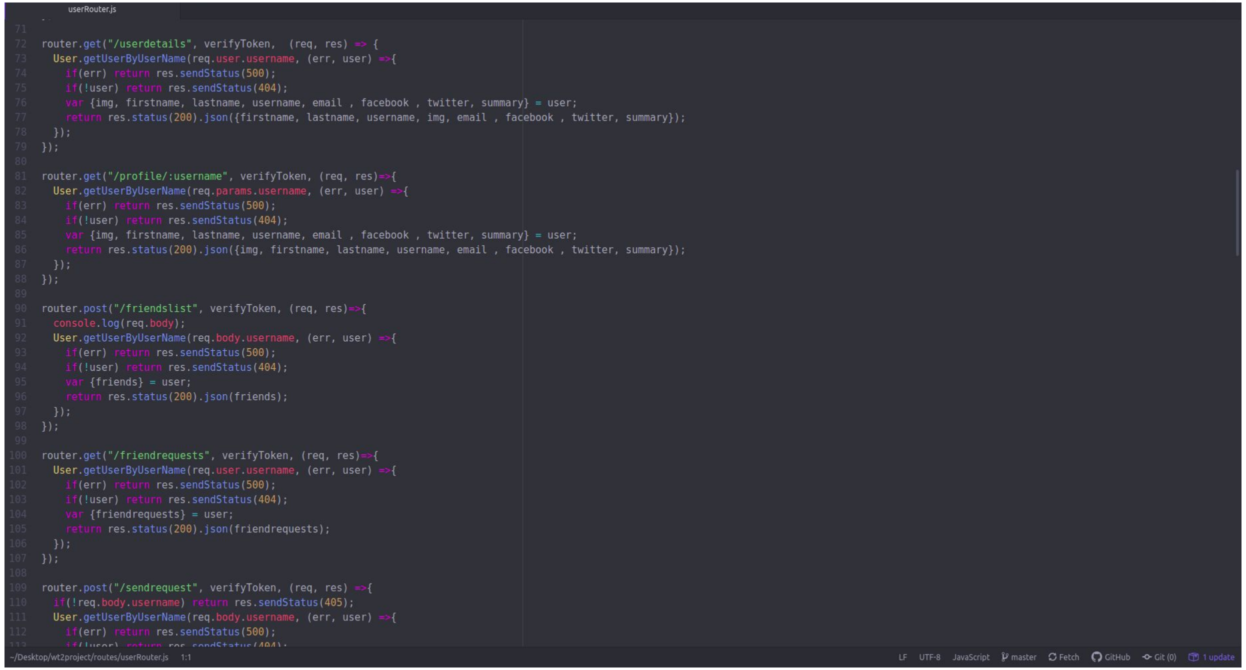
Support for let and class redeclarations
When experimenting with new code in the Console, repeating let or class declarations no longer causes errors.

Improved WebAssembly debugging
The Sources panel has increased support for stepping over code, setting breakpoints, and resolving stack traces in source languages.

Network panel updates

Techniques Implemented

- **REST API**: our nodejs backend application makes use of a stateless REST API that serves the required data to the client in JSON format.

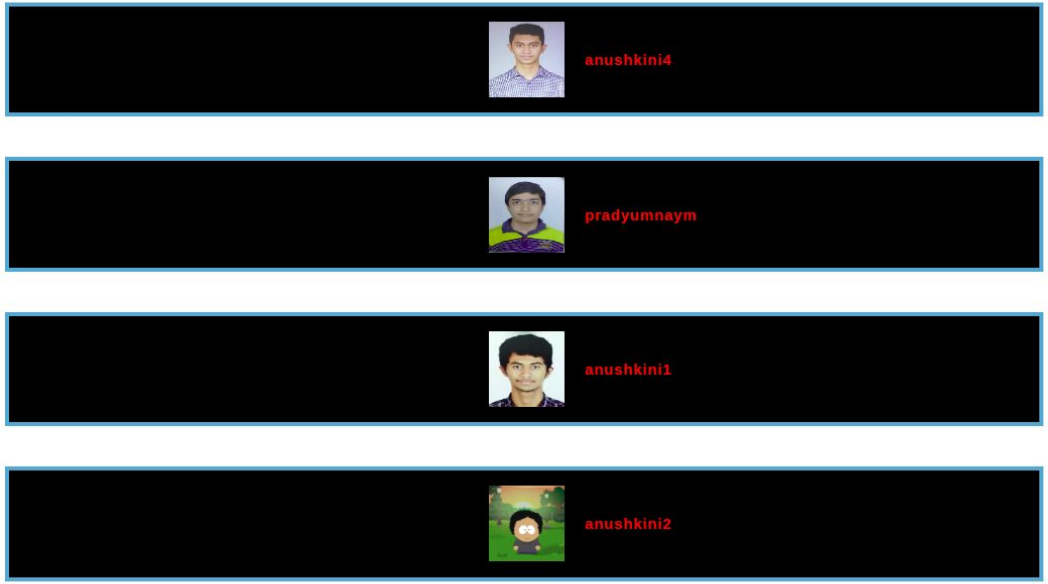


```javascript

router.get("/userdetails", verifyToken,  (req, res) => {
  User.getUserByUserName(req.user.username, (err, user) =>{
    if(err) return res.sendStatus(500);
    if(!user) return res.sendStatus(404);
    var {img, firstname, lastname, username, email , facebook , twitter, summary} = user;
    return res.status(200).json({firstname, lastname, username, img, email , facebook , twitter, summary});
  });
});

router.get("/profile/:username", verifyToken, (req, res)=>{
  User.getUserByUserName(req.params.username, (err, user) =>{
    if(err) return res.sendStatus(500);
    if(!user) return res.sendStatus(404);
    var {img, firstname, lastname, username, email , facebook , twitter, summary} = user;
    return res.status(200).json({img, firstname, lastname, username, email , facebook , twitter, summary});
  });
});

router.post("/friendslist", verifyToken, (req, res)=>{
  console.log(req.body);
  User.getUserByUserName(req.body.username, (err, user) =>{
    if(err) return res.sendStatus(500);
    if(!user) return res.sendStatus(404);
    var {friends} = user;
    return res.status(200).json(friends);
  });
});

router.get("/friendrequests", verifyToken, (req, res)=>{
  User.getUserByUserName(req.user.username, (err, user) =>{
    if(err) return res.sendStatus(500);
    if(!user) return res.sendStatus(404);
    var {friendrequests} = user;
    return res.status(200).json(friendrequests);
  });
});

router.post("/sendrequest", verifyToken, (req, res) =>{
  if(!req.body.username) return res.sendStatus(405);
  User.getUserByUserName(req.body.username, (err, user) =>{
    if(err) return res.sendStatus(500);
    if(!user) return res.sendStatus(404);
```

~/Desktop/wt2project/routes/userRouter.js 1:1
LF UTF-8 JavaScript master Fetch GitHub Git (0) 1 update

Intelligent Functionality

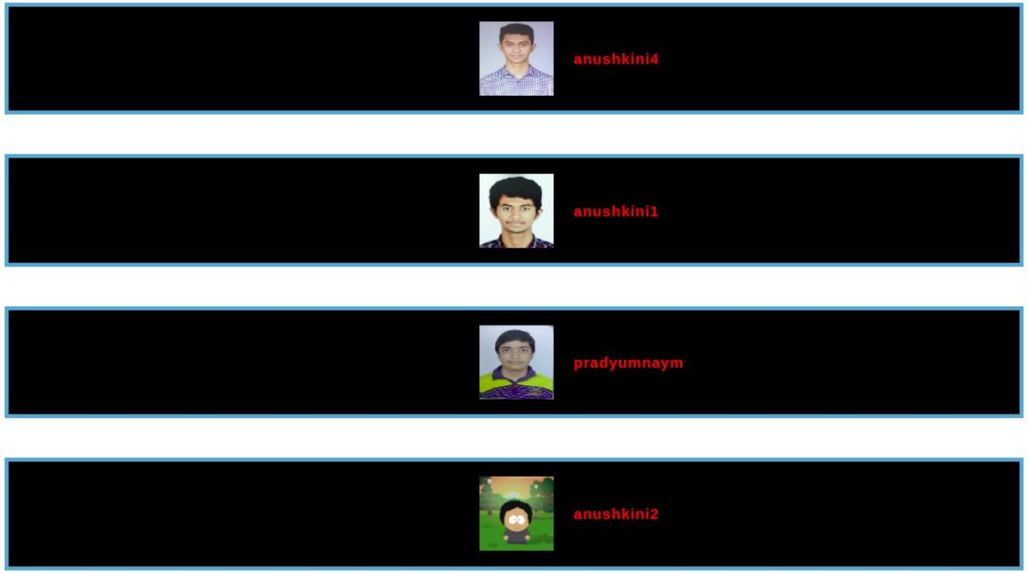
Smart friend reordering for profiles: We have implemented a convolutional neural network to extract features from users' profile pictures. Based on the euclidean distance of the features of the friend profile pictures, the friends list is ordered.


anushkini4
pradyumnaym
anushkini1
anushkini2

Initial Ordering

Intelligent Functionality

Smart friend reordering for profiles: We have implemented a convolutional neural network to extract features from users' profile pictures. Based on the euclidean distance of the features of the friend profile pictures, the friends list is ordered.

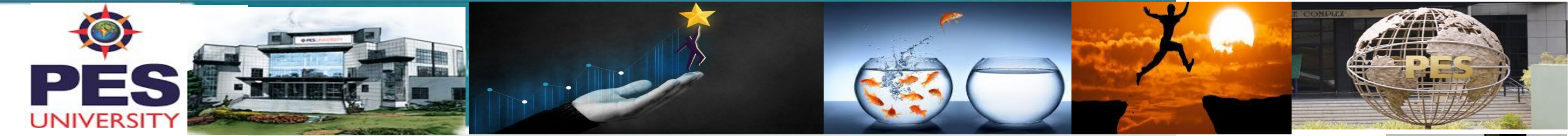

anushkini4

anushkini1

pradyumnaym

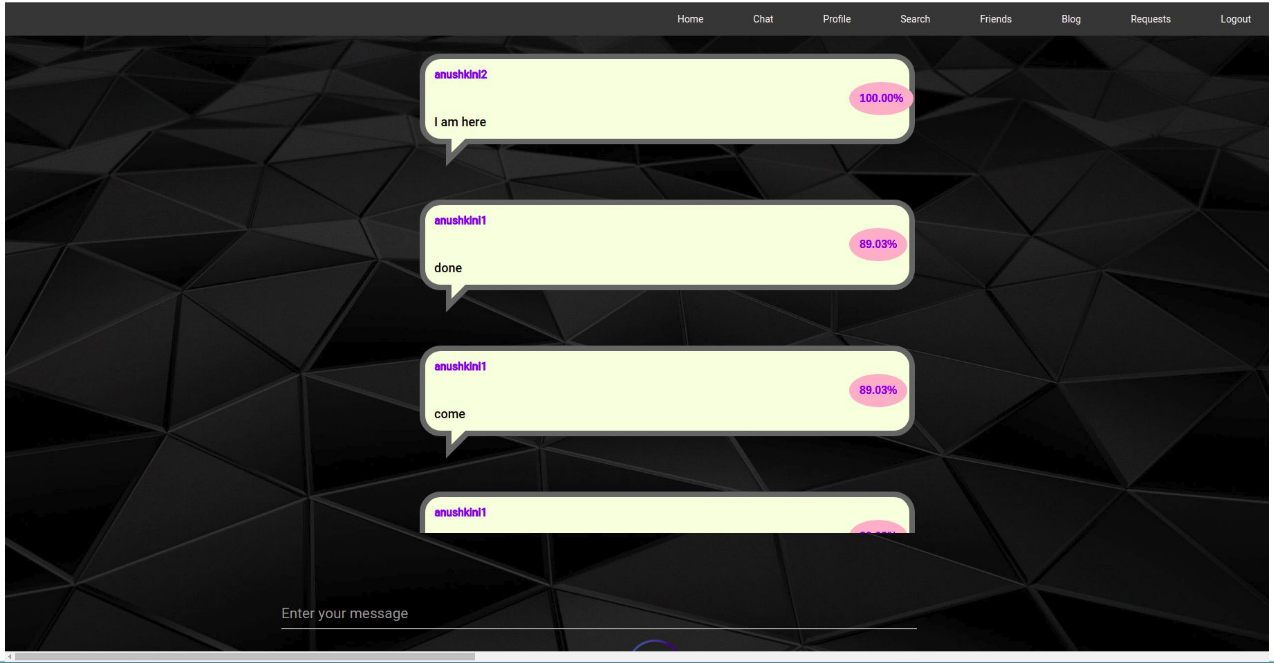
anushkini2

After running script



Intelligent Functionality

Synergy Indicator: We have developed a smart synergy indicator for the chat section of our website, which gives users a percentage of their synergy with another user. This can be used to make friends who play the same games as you and as a fun indicator of how similar you are to another user.



anushkini2
100.00%
I am here

anushkini1
89.03%
done

anushkini1
89.03%
come

anushkini1

Enter your message

Q / A