

ATHARVA EDUCATIONAL TRUST

ATHARVA COLLEGE OF ENGINEERING

MALAD, MUMBAI



A MINI PROJECT REPORT ON

“ONLINE GAMING WEBSITE”

SUBMITTED BY:

SAUNDARYA JUNJUR (29)
GOKUL KOMBATH (37)
AJAY KUSHWAHA (41)

UNDER THE GUIDANCE OF:

Prof. REEENA SOMANI

DEPARTMENT OF INFORMATION TECHNOLOGY

(2018-2019)

CERTIFICATE

This is to certify that the project entitled “**ONLINE GAMING WEBSITE**” is a bonafide work of **SAUNDARYA JUNJUR (29), GOKUL KOMBATH (37), AJAY KUSHWAHA (41)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “**TEIT**” in “**IP**”.

(GUIDE SIGNATURE)

Prof. Reena Somani

(HOD SIGNATURE)

Prof. Neelima Pathak

Dissertation Approval Certificate

This project report entitled website on ONLINE GAMING WEBSITE by SAUNDARYA JUNJUR, GOKUL KOMBATH, AJAY KUSHWAHA is approved for the degree of Bachelor of Engineering in Information Technology.

Examiners

1.-----

Name:-

Date:-

Place:-

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

SAUNDARYA JUNJUR(29)

GOKUL KOMBATH (37)

AJAY KUSHWAHA (41)

Date: 25-10-2018

INDEX

CHAPTER I

1.1 Introduction	1
1.2 Problem statement	1
1.3 Scope and Aim	2
1.4 Objective	2

CHAPTER II

2.1 Hardware and Software Requirements	3
--	---

CHAPTER III

3.1 Code	4
3.2 Snapshots	9

CHAPTER IV

4.1 Conclusion	13
----------------	----

LIST OF FIGURES

3.2.1	Welcome Page	9
3.2.2	Arcade Category Page	9
3.2.3	Sign In Page	10
3.2.4	Sign Up Page	10
3.2.5	Main Page	11
3.2.6	Admin Page	11
3.2.7	Gaming Page	12
3.2.8	Strategy Category Page	12

CHAPTER I

1.1 INTRODUCTION

Online gaming is hugely popular with children and young people. Annual research conducted by OFCOM shows that gaming is still one of the top activities enjoyed by 5-16 year olds online, with many of them gaming via mobile devices and going online using their games console. From sport related games to mission based games and quests inspiring users to complete challenges, interactive games cater for a wide range of interests, and can enable users to link up and play together. Most games now have an online element to them; allowing users to take part in leader boards, join group games or chat to others. Internet connectivity in a game adds a new opportunity for gamers as it allows players to find and play against, or with, other players. These may be their friends or family members or even other users in the game from around the world (in a multiplayer game). We know that parents and care-takers do have questions and concerns about games, often about the type of games their child plays, who they may be speaking to and for how much time their child is playing. This leaflet provides an introduction to online gaming and advice for parents specifically related to gaming.

The main goal of our online gaming website is to provide entertainment to people without having to download the games. It will also help to interact with people around the world. This website provides a wide variety of games which will help the players improve their coordination, problem-solving skills, memory, attention and concentration etc. It also provides a great source of learning as there are many educational related games.

1.2 PROBLEM STATEMENT

The problem with offline games is that one has to download the game to play. While it may seem convenient sometimes, but when there is a need to play variety of games then it becomes too hectic to download all the games individually. It may also happen that some may end up downloading spam, viruses, malicious software etc. In our website people can have fun by playing the games without downloading. All they need is a good internet connection.

1.3 SCOPE AND AIM

SCOPE: Our project consists of variety of games such as arcade, action, educational, puzzles etc. A person who wish to play all the games in one particular website then our website is preferable. It overcomes the tedious problem of downloading all the games from different websites. Online games play a very important in developing time management. Can be used anywhere any time as it is a web based application

AIM: This project is aimed to design and develop online games with vast varieties of games without downloading all the games from all the websites. Studies say that playing online games improves many skills like concentration, memory etc.

1.4 OBJECTIVE

The main objective is to develop an online gaming website is

- To play all variant of games in single website without downloading.
- To improve memory concentration, user friendly interaction etc.
- To have fun and enjoy all the games.

CHAPTER II

2.1 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware requirements	Software requirements	Other Requirements
<p>We strongly recommend a computer fewer than 5 years old.</p> <ol style="list-style-type: none">1. Processor: Intel Core i3/i5/i72. Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)3. Hard Drive: Minimum 32 GB; Recommended 64 GB or more4. Memory (RAM): Minimum 1 GB; Recommended 2 GB or above5. Sound card w/speakers	<ol style="list-style-type: none">1. Windows, Linux2. HTML5 compatible web browsers(Chrome,Firefox,Safari,Microsoft Edge etc)3. Editors like Notepad++,Sublime,Editplus etc.4. Pycharm IDE , Python IDLE5. Flash Player	<ul style="list-style-type: none">• Internet Connection :Minimum 1mbps speed• Web Frameworks

CHAPTER III

3.1 CODE

Database Model:

```
from django.db import models
from registration.models import UserDetails

class Game(models.Model):
    game_images =
models.CharField(max_length=100)
    game_swf = models.CharField(max_length=100)
    game_type = models.CharField(max_length=100)
    game_rating = models.IntegerField(max_length=5)
    comment = models.CharField(max_length=300)
    game_display_name =
models.CharField(max_length=100)
    class Meta:
        db_table = "Game_Details"

class Comments(models.Model):
    comment = models.CharField(max_length=300)
    game_name = models.CharField(max_length=40)
    Username = models.CharField(max_length=40)

    class Meta:
        db_table = "Comments"

from django.db import models

class UserDetails(models.Model):
    Username = models.CharField(max_length=20)
    Email = models.CharField(max_length=20)
    Password = models.CharField(max_length=20)
    Avatar_image =
models.CharField(max_length=50)

    class Meta:
        db_table = "User_Details"

class Avatars(models.Model):
    avatar_image_name =
models.CharField(max_length=50)

    class Meta:
        db_table = "avatar"
```

URL:

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', welcome, name='welcome'),
    path('signin/', sign_in, name='signin'),
    path('signin/redirect', auth_user, name='signin'),
    path('signup/', registration, name='signup'),
    path('signup/create_user', create_user,
name='signup'),
    path('mainpage/', mainpage, name='mainpage'),
    path('arcade/', arcade, name='arcade'),
    path('action/', action, name='action'),
    path('puzzle/', puzzle, name='puzzle'),
    path('sport/', sport, name='sport'),
    path('strategy/', strategy, name='strategy'),
    path('mainpage/show_game/<game_name>.swf',
show_game, name='show_game'),
    path('mainpage/search/', search, name='search'),
    path('mainpage/logout/', logout, name='logout'),
    path('mainpage/setting/add_a_game/', addGame,
name='addGame'),
    path('mainpage/setting/delete_a_game/',
removeGame, name='removeGame'),

    path('mainpage/setting/add_a_game/gotoApproval',
, gotoApproval, name='gotoApproval'),
    path('mainpage/mycomment/', making_comment,
name='logout'),
    path('mainpage/top_chart/<chart_name>',
top_chart, name='top_chart'),
    path('mainpage/filter/<filter_name>',
filter_type, name='filter_type'),
    path('game/<game_name>.swf',
welcome_game_play,
name='welcome_game_play'),
]
```

Views:

```
def mainpage(request):
    objects = Game.objects.all()
    paginator = Paginator(objects, 24)
    page = request.GET.get('page')
    objects = paginator.get_page(page)

    try:
        uname = request.session['UserName']
```

```
        user_images =
        UserDetails.objects.filter(Username=uname)
        myimage = ""
        for images in user_images:
            myimage = images.Avatar_image
        return render(request, 'mainpage.html',
{"mygame": objects, "username": uname,
"avatar_image": myimage})
    except:
        return redirect("http://127.0.0.1:8000/")

def show_game(request, game_name):
    comment_user_image = ""
    try:
        uname = request.session['UserName']
        user_images =
        UserDetails.objects.filter(Username=uname)
        myimage = ""
        for images in user_images:
            myimage = images.Avatar_image
        comment_whole_details =
        Comments.objects.filter(game_name=game_name).re
verse()
        for my_comment in comment_whole_details:
            comment_user_details =
            UserDetails.objects.filter(Username=my_comment.U
sername)
            for user in comment_user_details:
                comment_user_image = user.Avatar_image
            my_comment.avatar_image =
            comment_user_image
        return render(request, 'show_games.html',
{"game_title": game_name,
'username': uname, 'avatar_image':
myimage, 'comment':comment_whole_details})
    except:
        return redirect("http://127.0.0.1:8000/")
```

```
def filter_type(request, filter_name):
    objects =
    Game.objects.filter(game_type=filter_name)
    paginator = Paginator(objects, 24)
    page = request.GET.get('page')
    objects = paginator.get_page(page)

    try:
        uname = request.session['UserName']
        user_images =
```

```

UserDetails.objects.filter(Username=uname)
myimage = ""
for images in user_images:
    myimage = images.Avatar_image
if filter_name == "Arcade":
    return render(request, 'mainpage.html',
        {"mygame": objects, "active2":
'active', "username": uname, 'avatar_image':
myimage})
    elif filter_name == "Action":
        return render(request, 'mainpage.html',
            {"mygame": objects, "active3":
'active', "username": uname, 'avatar_image':
myimage})
    elif filter_name == "Puzzle":
        return render(request, 'mainpage.html',
            {"mygame": objects, "active4":
'active', "username": uname, 'avatar_image':
myimage})
    elif filter_name == "Sport":
        return render(request, 'mainpage.html',
            {"mygame": objects, "active5":
'active', "username": uname, 'avatar_image':
myimage})
    elif filter_name == "Strategy":
        return render(request, 'mainpage.html',
            {"mygame": objects, "active6":
'active', "username": uname, 'avatar_image':
myimage})
    else:
        return HttpResponseNotFound("No such
category")
except:
    return redirect("http://127.0.0.1:8000/")

def search(request):
    game_name_contains = request.GET.get('term')
    objects =
Game.objects.filter(game_display_name__icontains=
game_name_contains)
    try:
        uname = request.session['UserName']
        user_images =
UserDetails.objects.filter(Username=uname)
        myimage = ""
        for images in user_images:
            myimage = images.Avatar_image
        return render(request, 'mainpage.html',

```

```

        {"mygame": objects, "search_term":
game_name_contains, "username": uname,
        'avatar_image': myimage})

    except:
        return redirect("http://127.0.0.1:8000/")

def top_chart(request, chart_name):
    try:
        uname = request.session['UserName']
        user_images =
UserDetails.objects.filter(Username=uname)
        myimage = ""
        for images in user_images:
            myimage = images.Avatar_image
        if chart_name == "t_10_arcade":
            objects =
Game.objects.filter(game_type="Arcade").order_by(
            '-game_rating')[:10]
            return render(request, 'mainpage.html',
                {"mygame": objects, "username": uname,
                'avatar_image': myimage})

        elif chart_name == "t_10_action":
            objects =
Game.objects.filter(game_type="Action").order_by(
            '-game_rating')[:10]
            return render(request, 'mainpage.html',
                {"mygame": objects, "username": uname,
                'avatar_image': myimage})

        elif chart_name == "t_10_puzzle":
            objects =
Game.objects.filter(game_type="Puzzle").order_by(
            '-game_rating')[:10]
            return render(request, 'mainpage.html',
                {"mygame": objects, "username": uname,
                'avatar_image': myimage})

        elif chart_name == "t_10_sport":
            objects =
Game.objects.filter(game_type="Sport").order_by(
            '-game_rating')[:10]
            return render(request, 'mainpage.html',
                {"mygame": objects, "username": uname,
                'avatar_image': myimage})

        elif chart_name == "t_10_strategy":

```

```

        objects =
Game.objects.filter(game_type="Strategy").order_b
y('-game_rating')[:10]
        return render(request, 'mainpage.html',
{"mygame": objects, "username": uname,
'avatar_image': myimage})

        elif chart_name == "t_10_editors_choice":
            objects = Game.objects.filter().order_by('-
game_rating')[:10]
            return render(request, 'mainpage.html',
{"mygame": objects, "username": uname,
'avatar_image': myimage})

        elif chart_name == "all_games":
            objects = Game.objects.all()
            paginator = Paginator(objects, 24)
            page = request.GET.get('page')
            objects = paginator.get_page(page)
            return render(request, 'mainpage.html',
{"mygame": objects, "username": uname,
'avatar_image': myimage})

    else:
        return HttpResponseRedirect("No such

```

```

def welcome(request):
    return render(request,
'welcome.html',{'active1' : 'active'})

def arcade(request):
    objects =
Game.objects.filter(game_type='Arcade')
    return render(request,
'type_game_show.html',{'mygame' :
objects,"active2" : 'active'})

def action(request):
    objects = Game.objects.filter(game_type='Action')
    return render(request,
'type_game_show.html',{'mygame' :
objects,"active3" : 'active'})

def puzzle(request):
    objects = Game.objects.filter(game_type='Puzzle')
    return render(request,
'type_game_show.html',{'mygame' :

```

```

category")

except:
    return redirect("http://127.0.0.1:8000/")

def making_comment(request):
    comment = request.POST.get('comment', "")
    game_name = request.POST.get('game_name',
'')[20:-4]
    user_name = request.POST.get('user_name', "")
    dr1 = Comments(comment=comment,
game_name=game_name, Username=user_name)
    dr1.save()
    return JsonResponse({'success': 'added the
comment'})

def logout(request):
    try:
        del request.session['UserName']
        return redirect("http://127.0.0.1:8000/")
    except KeyError:
        pass
    return HttpResponseRedirect("You're logged out.")

```

```

objects,"active4" : 'active'})

def sport(request):
    objects = Game.objects.filter(game_type='Sport')
    print(objects)
    return render(request,
'type_game_show.html',{'mygame' :
objects,"active5" : 'active'})

def strategy(request):
    objects =
Game.objects.filter(game_type='Strategy')
    return render(request,
'type_game_show.html',{'mygame' :
objects,"active6" : 'active'})

def welcome_game_play(request, game_name):
    return render(request, 'game_play.html',
{"game_title": game_name})

```

```

def registration(request):
    all_avatar = Avatars.objects.all()

    avatar_display = []
    for i in range(0, 11):
        temp = []
        j = 0
        for avatar in all_avatar:
            temp_avatar = copy.deepcopy(avatar)
            if i == j:
                temp_avatar.display = True
            else:
                temp_avatar.display = False
            temp.append(temp_avatar)
            j += 1
        avatar_display.append(temp)

    return render(request, 'registration.html',
    {'all_avatar': avatar_display, 'avatars': all_avatar})

def create_user(request):
    uname = request.POST.get('Username', '')
    eml = request.POST.get('Email', '')
    pwd = request.POST.get('Password', '')
    image_name = request.POST.get('Imagename', '')
    image_name = image_name[8:]
    dr1 = UserDetails(Username=uname, Email=eml,
    Password=pwd, Avatar_image=image_name)
    dr1.save()
    request.session['UserName']=uname

```

```

    return
    redirect("http://127.0.0.1:8000/mainpage/")

def addGame(request):
    return render(request, 'settings.html')

def gotoApproval(request):
    game_name = request.POST.get('game_name', '')
    game_type = request.POST.get('game_type', '')
    game_rating = request.POST.get('game_rating',
    '')
    game_image = request.FILES['image_file']
    game_swf = request.FILES['swf_file']
    fs = FileSystemStorage()
    file_image = fs.save(game_image.name,
    game_image)
    file_swf = fs.save(game_swf.name, game_swf)
    dr1 = Game(game_images=file_image,
    game_swf=file_swf, game_type=game_type,
    game_rating=game_rating, comment="",
    game_display_name=game_name)
    dr1.save()
    return
    redirect("http://127.0.0.1:8000/mainpage/setting/ad
    d_a_game/")

def removeGame(request):
    objects = Game.objects.all()
    return render(request, 'admin_control
    .html', {"all_game" : objects})

```

3.2 SNAPSHOTS

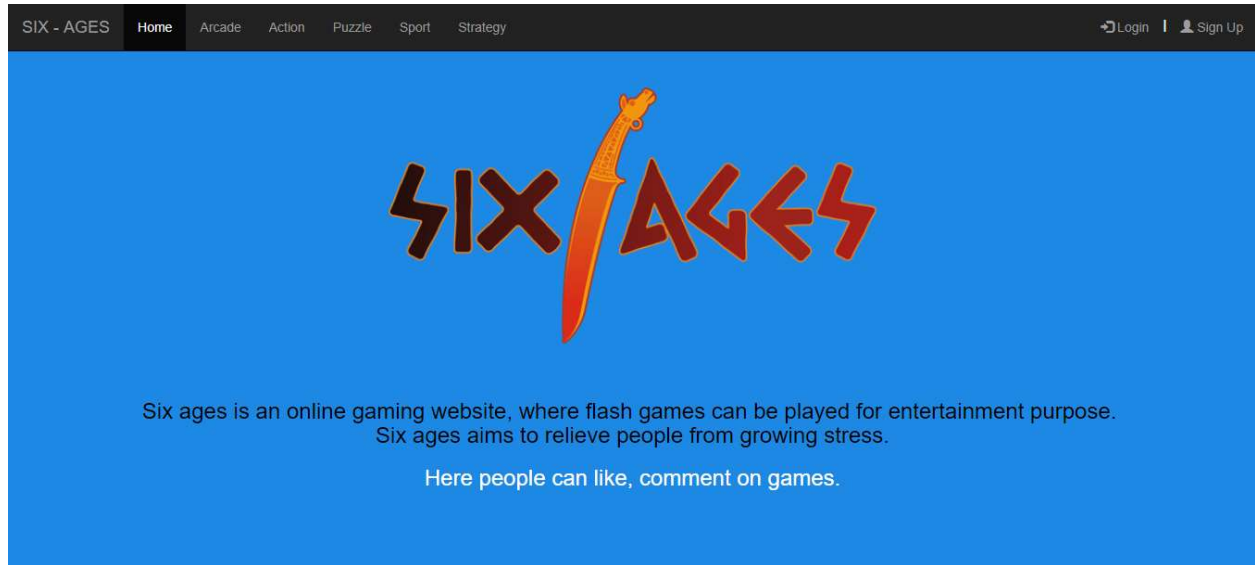


Fig : 3.2.1 Welcome Page

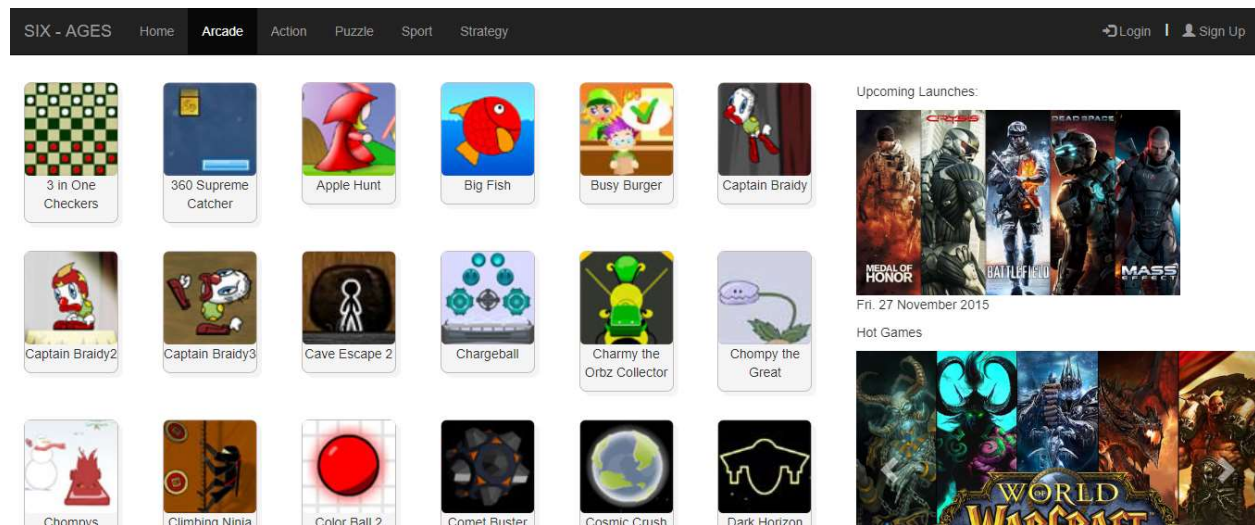


Fig : 3.2.2 Arcade Category Page

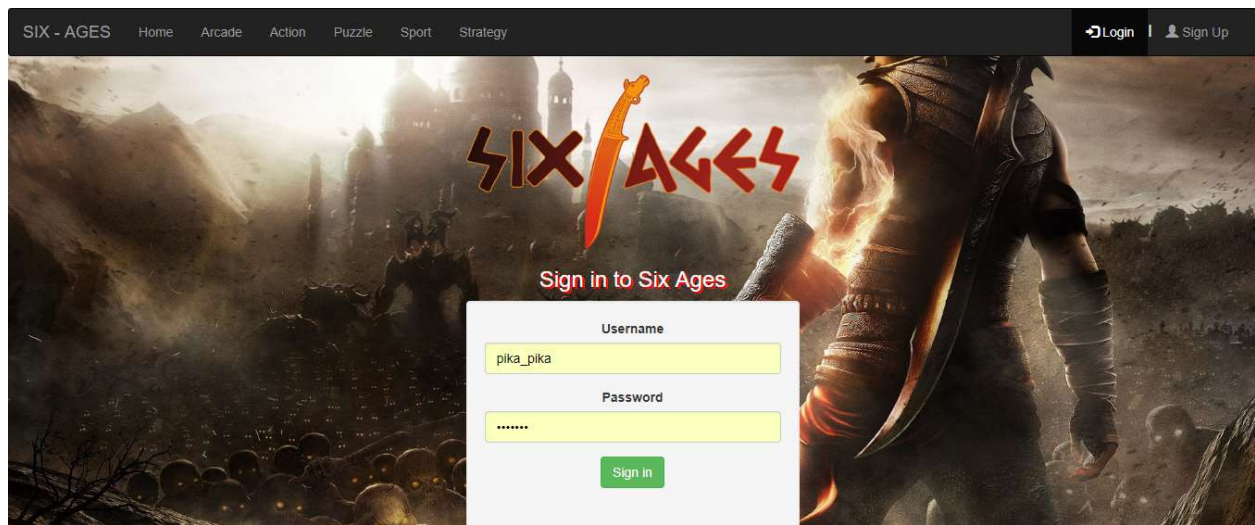


Fig : 3.2.3 Sign In

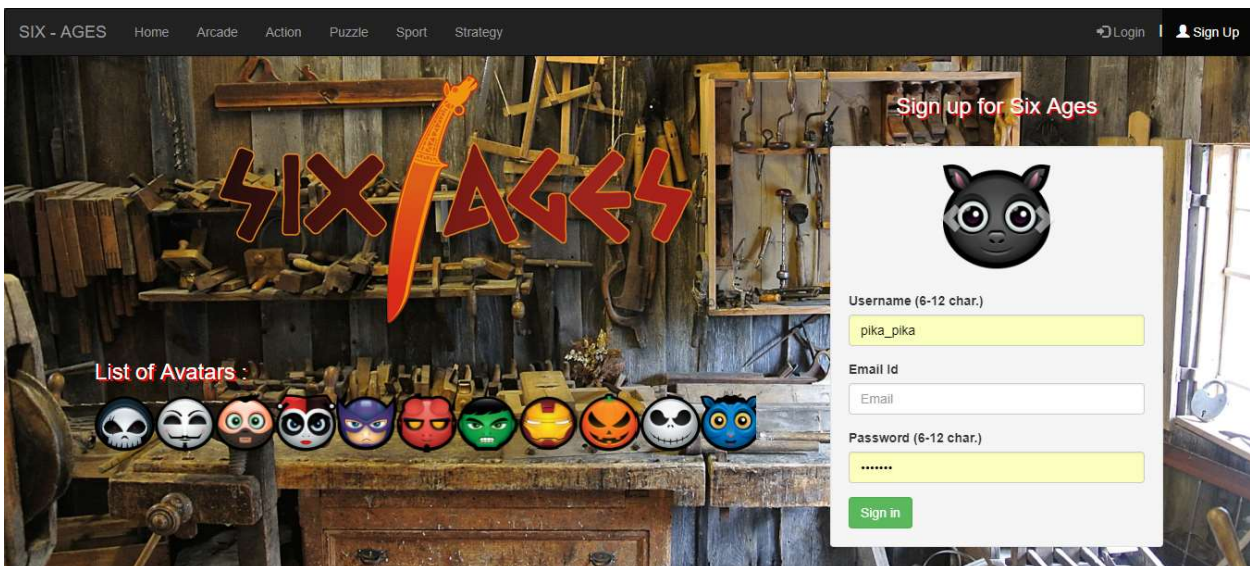


Fig : 3.2.4 Sign Up

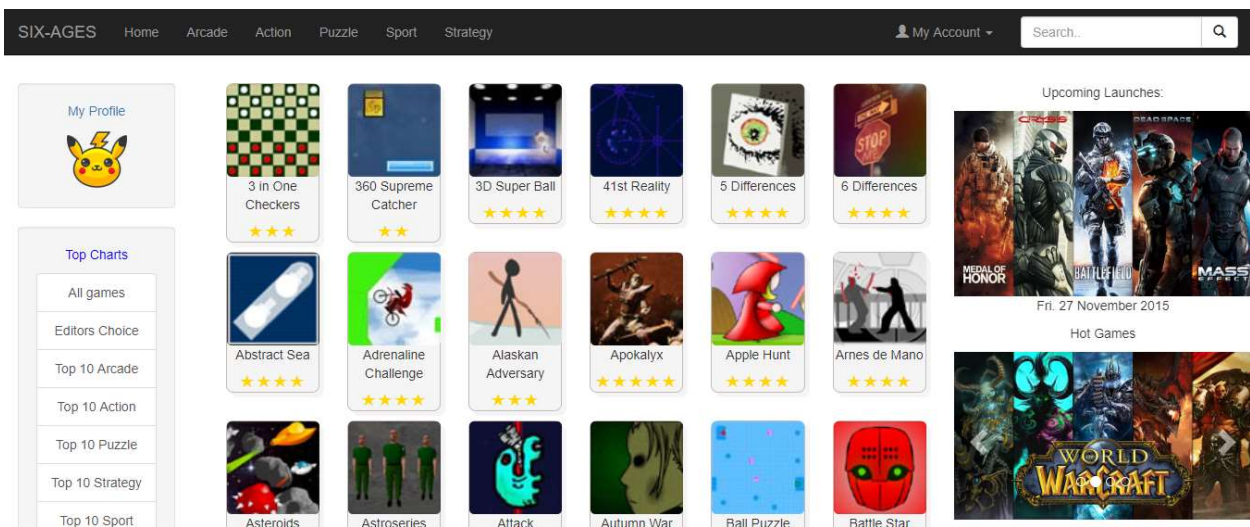


Fig : 3.2.5 Main Page

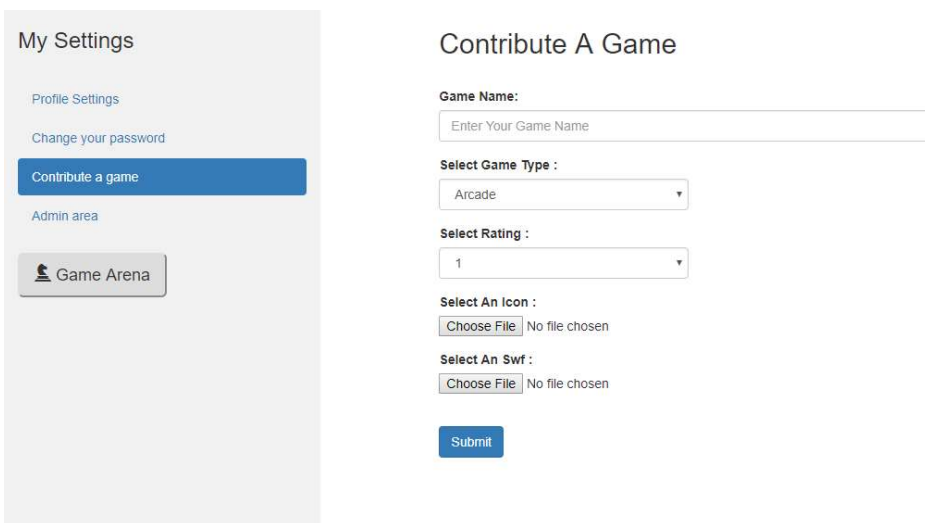


Fig : 3.2.6 Admin Page

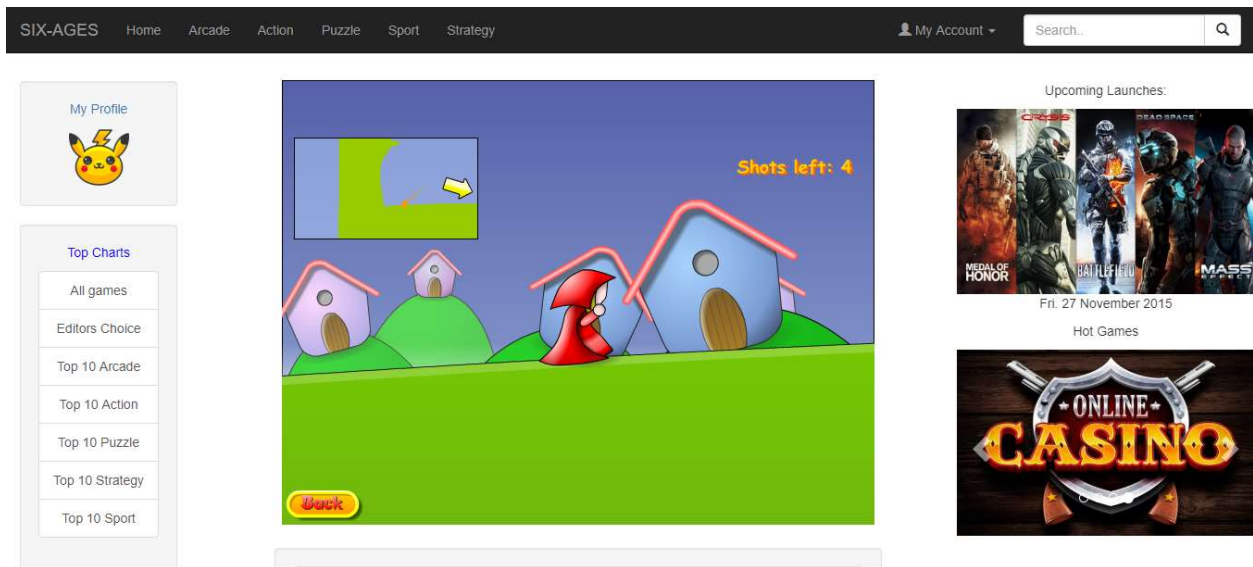


Fig : 3.2.7 Gaming Page

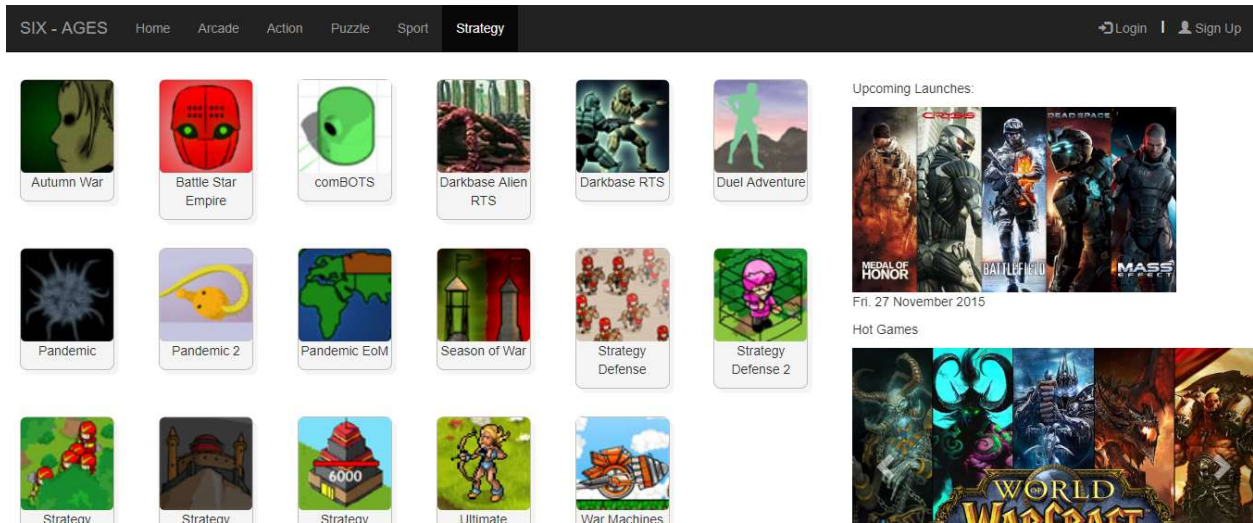


Fig : 3.2.8 Strategy Category Page

CHAPTER IV

4.1 CONCLUSION

Although there are many cons, we have concluded as a group that the pros heavily outweigh them, online gaming helps the development of children, provides a fun and exciting way to pass time and develops teamwork skills, they enhance skills such as technology skills that are required in today's life which is constantly changing and growing.

But we do agree that online gaming does have a downside like everything in life, and that use should be limited and online gamers should be aware of the things that can possibly happen if their time gaming is abused. Parents should limit the time their children are allowed to play online, and only allow use if all homework and other commitments are completed. Gaming should be a reward not a given.

REFERENCES

- [1] www.w3schools.com.
- [2] www.quora.com
- [3] www.github.com
- [4] <https://www.youtube.com/playlist?list=PL6gx4CwI9DGBImzzFcLgDhKTTfNLfX1IK>