**Web Programming Lab**

**MIDTERM Assignment Report**

2018314630

NaYeon Kim

**1. The design or structure of your homepage should be match with your topic.**

제 토픽은 콘솔 게임 회사, Explorer Games의 웹 페이지입니다. MainPage는 아래 사진(Figure 1)과 같으며 본사의 유명한 게임들을 소개하고 있습니다.

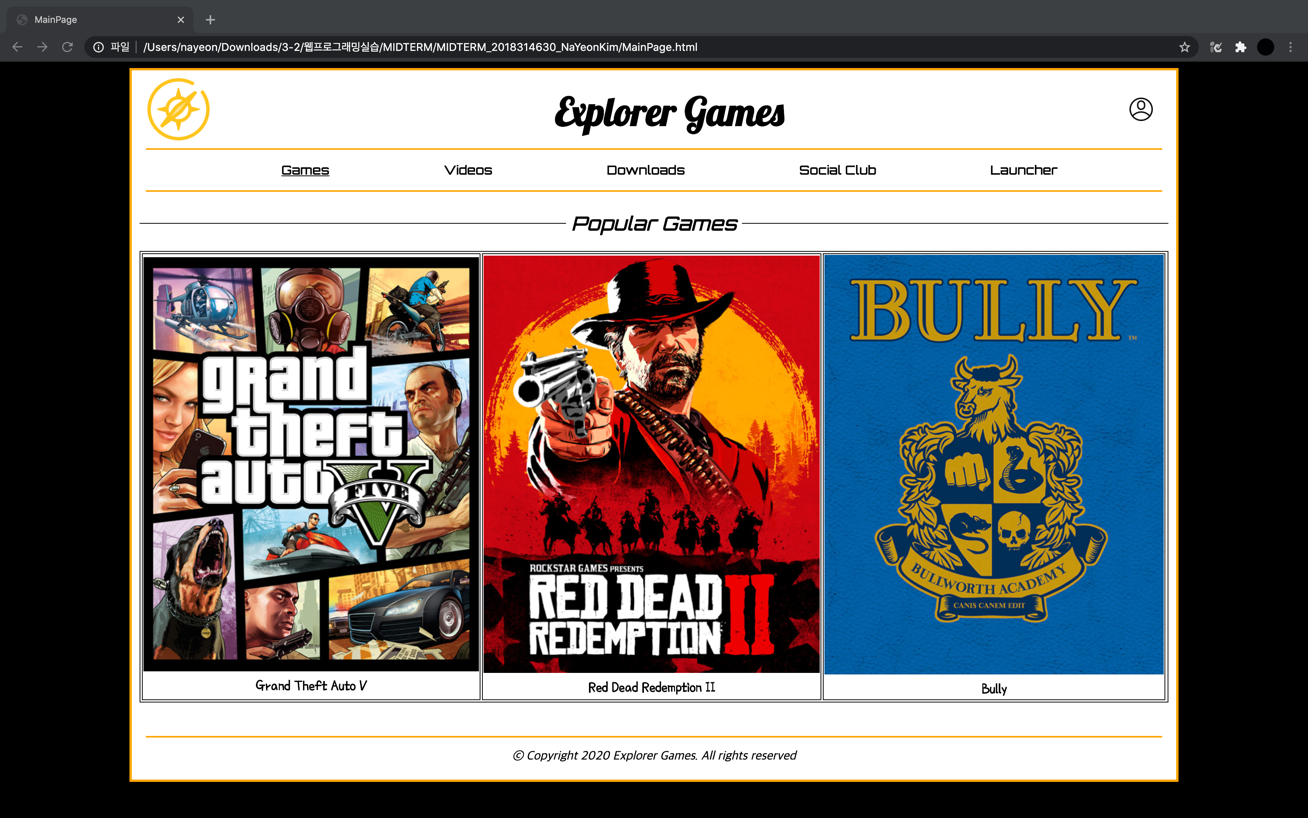


Figure 1: MainPage

**2. Make more than 3 html files, and the three html files should be able to move meaningfully to each other with links.**

제 프로젝트는 총 세 가지 html 파일들이 존재합니다. 위 사진의 MainPage를 나타내는 MainPage.html, 각 게임들의 하이라이트 영상들을 보여주는 VideoPage.html, 사용자의 계정 생성을 위한 SignUpPage.html입니다. 아래는 각각의 페이지들이 어떻게 연결되어 있는지를 보여줍니다. (Figure 2)

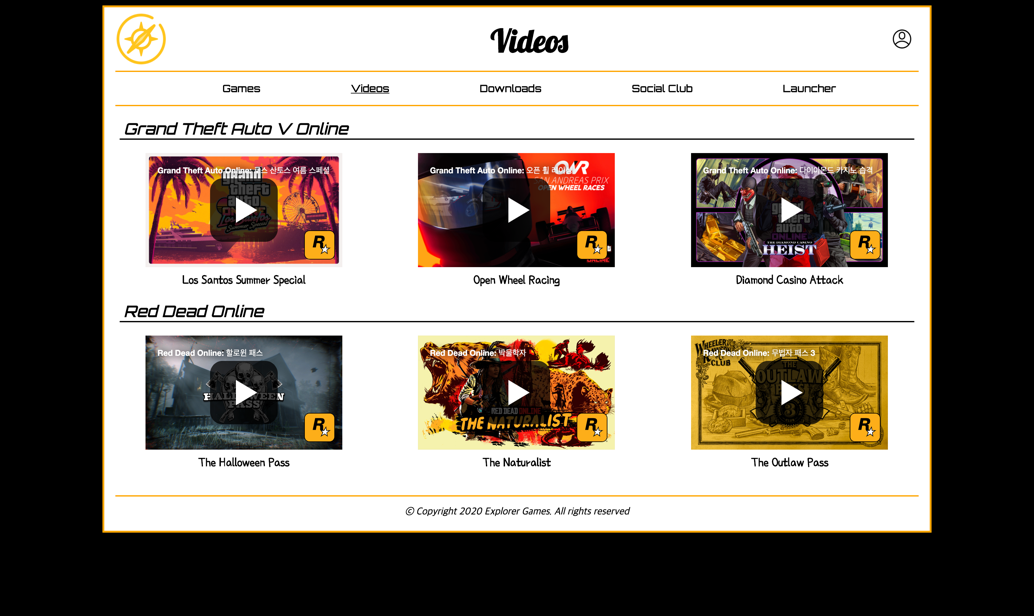
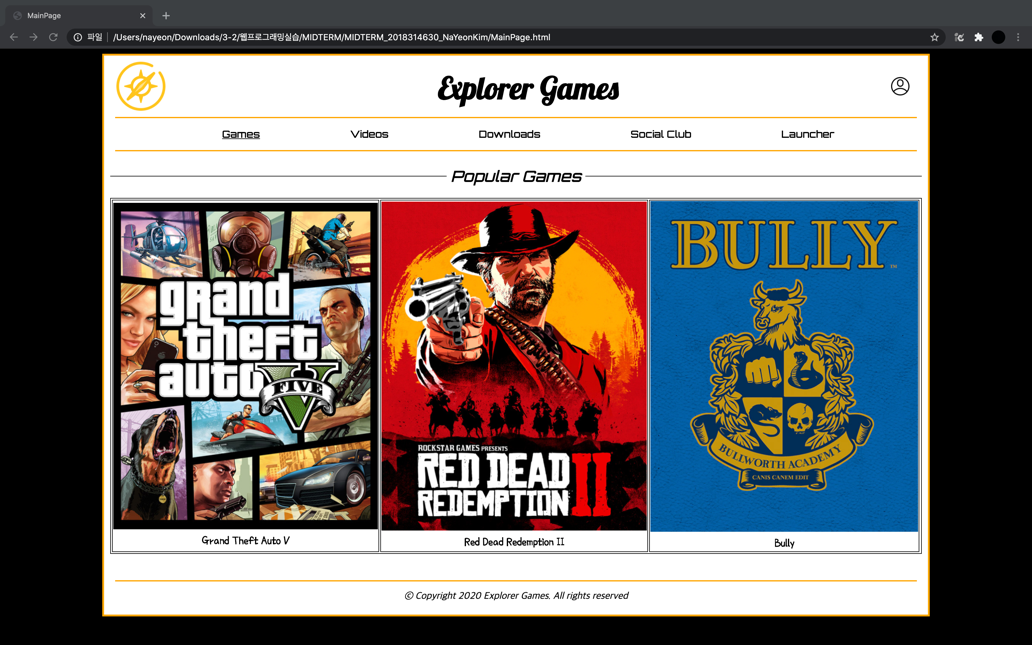


Figure 2: Links between html Pages

**3. Keep the basic form header, nav, content, footer.**

각각의 html 파일 모두 크게는 wrapper이 있고 그 하위에는 header, nav(SignUpPage 제외), content, footer 형식을 포함합니다.

**4. Do not use frameworks. (boostrap, wordpress, etc.)**

오로지 html과 css의 기능만으로 구현하였습니다.

**5. Make the page that collects user information using form tag. (radio boxes, check boxes)**

이는 SignUpPage.html의 form tag로 구현되어 있습니다. (Figure3, 4)

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 3: Radio boxes and Check boxes in SIgnUpPage.

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 4: Several input tags in form tag.

**6. Create more than 3 external css files.**

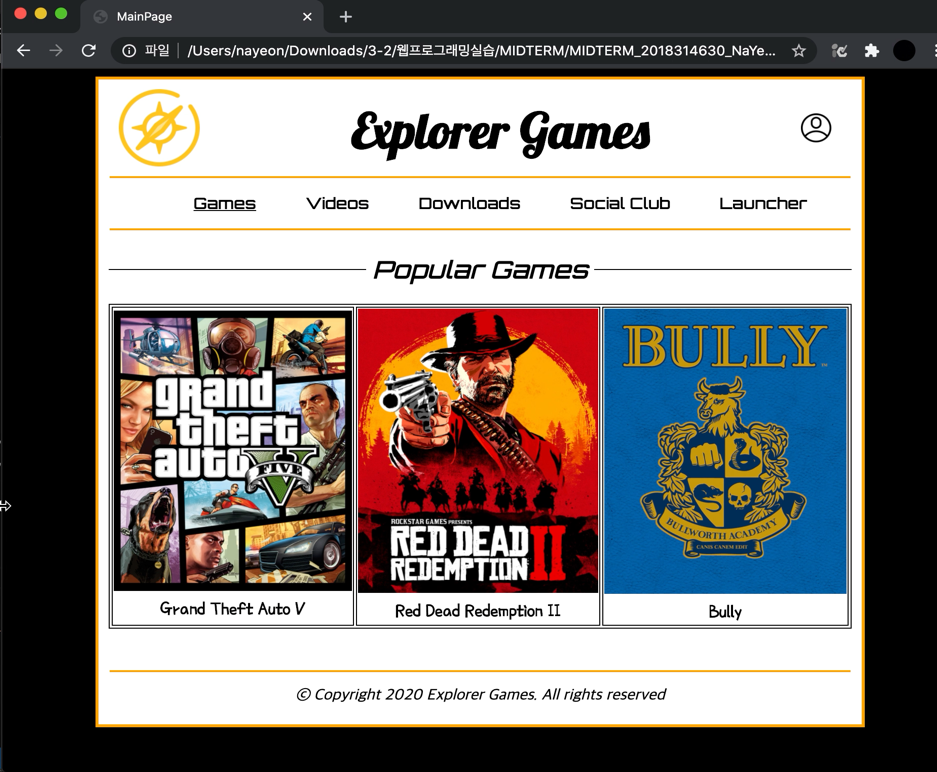
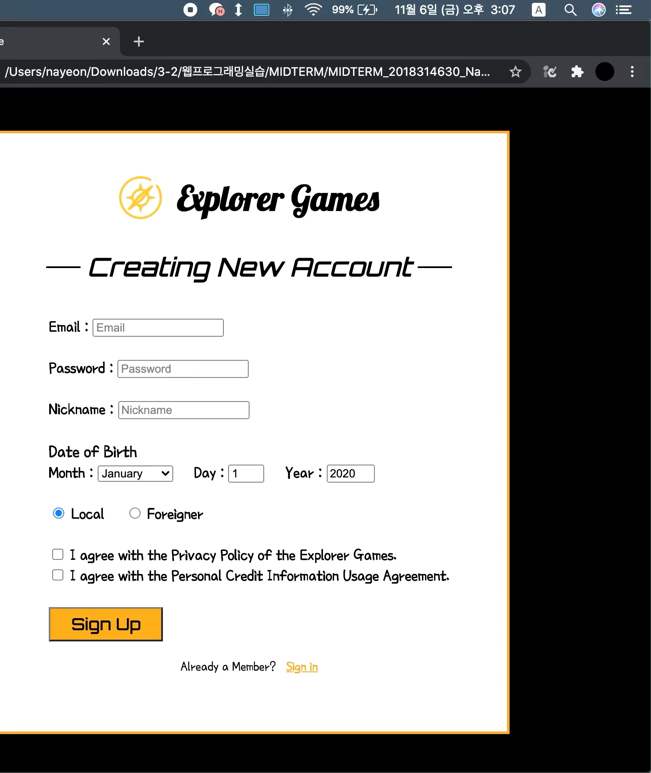
각각의 MainPage, VideoPage, SignUpPage는 모두 자신의 external css 파일을 갖습니다. (Figure 5)

텍스트이(가) 표시된 사진

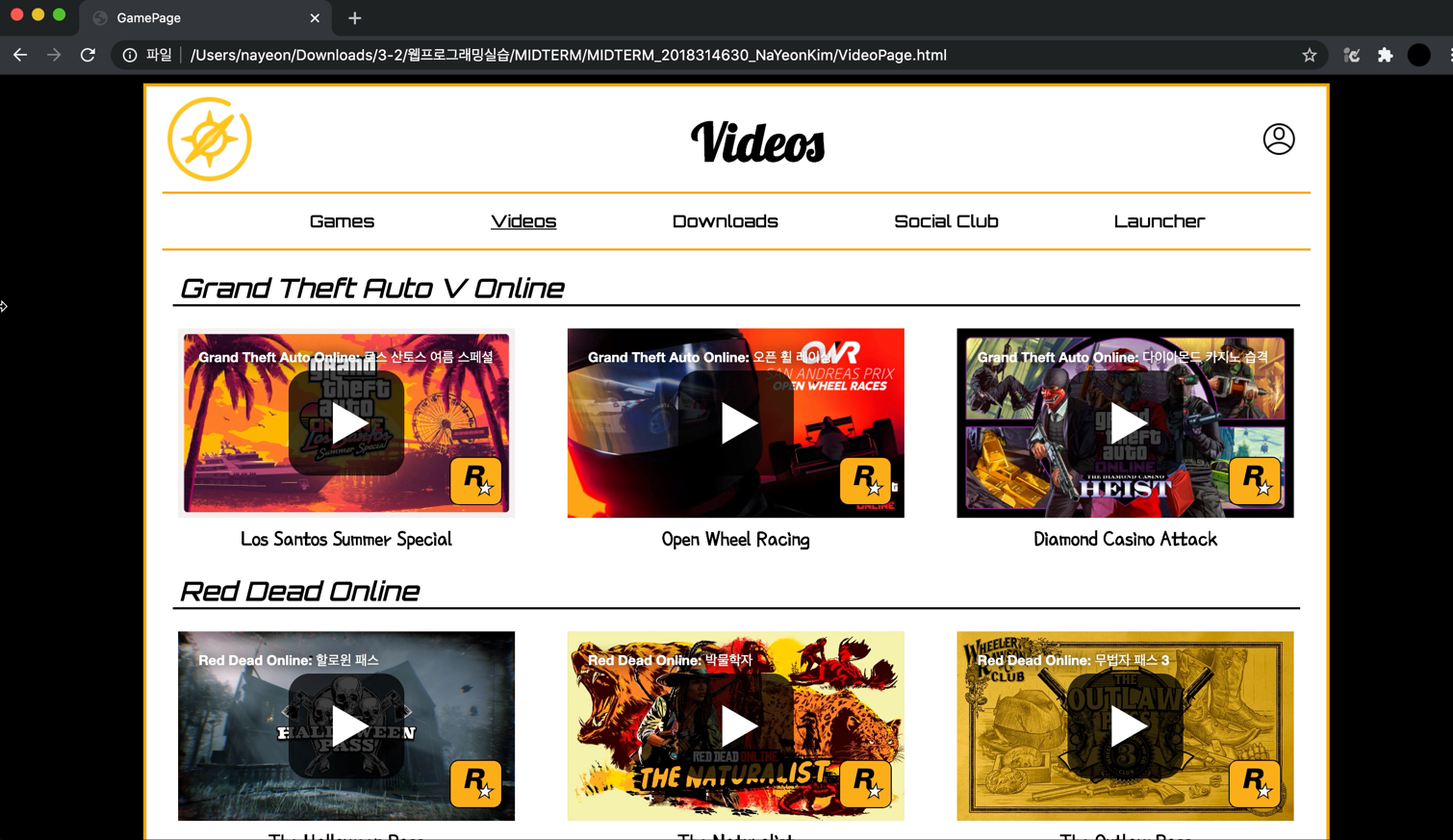
자동 생성된 설명

Figure 5: All files in my project

**7. Construct a website based on reactive.**

****아래 동영상을 세 개를 첨부하였습니다. (Video 1, 2, 3) 세 html 웹 페이지 모두 화면의 크기에 따라 다른 레이아웃을 보여주는 것을 알 수 있습니다. @media를 이용하여 max-width에 따라 Flex-direction을 바꾸거나 폰트의 크기를 조절하였습니다.

Video 1: SignUpPage Video 2: MainPage

Video 3 : VideoPage

**8. Homepage should include more than 2 multimedium.**

MainPage에 게임 image(img tag 사용, Figure 6), 그리고 VideoPage에 video(iframe tag 사용, Figure 7)가 첨부되어 있습니다.

**9. Use text-style to make it pretty.**

각 페이지의 텍스트들에 bold, italic 등의 font-weight 및 font-style을 추가했습니다.

**10. Use table tag.**

MainPage와 VideoPage에 이미지와 영상을 table로 정렬하였습니다. (Figure 6, 7)

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 6: Img tags in table tag

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 7: Iframe tags in table tag

**11. Create effective pages by using more than 3 web fonts such as provided from Google fonts.**

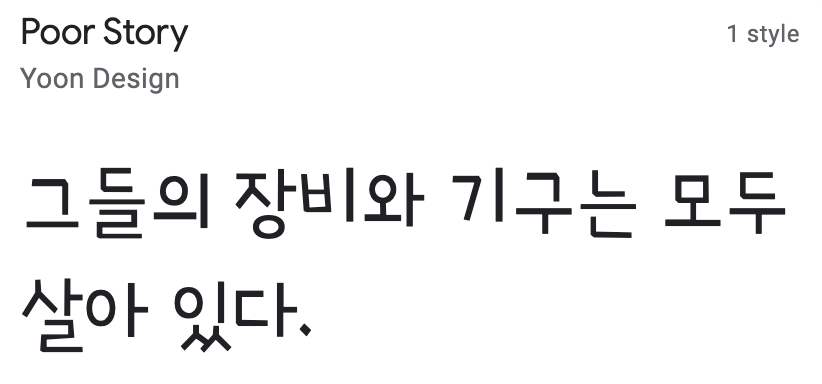
Lobster, Orbitron, Poor Story 이렇게 세 가지의 web font를 사용하였습니다. (Figure 8,9,10,11)

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

Figure 8: Orbitron Figure 9: Lobster

 Figure 10: Poor Story

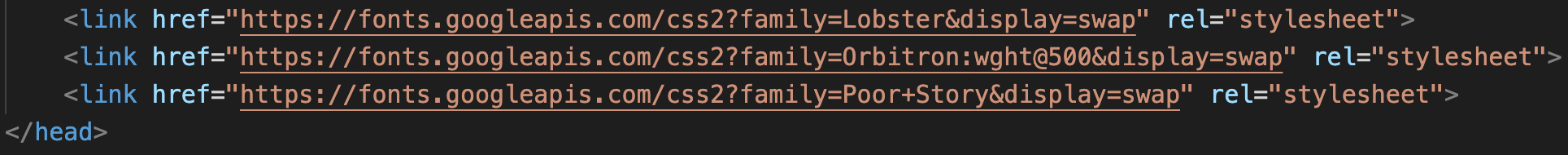
****

Figure 11: link with web fonts

**12. Use the flex property to create flex box containers.**

세 css 파일 모두 적절하게 flex property를 사용하였습니다. 모든 파일의 wrapper과 그 하위의Header, nav, content 클래스들의 정렬이 용이하도록 모두 flex를 적용하였습니다.

**13. Use transform property.**

Mainpage에서 각각의 이미지에 hover 효과를 주어 transform의 scale 값이 바뀌도록 하였습니다. (Figure 12)

또한, MainPage와 VideoPage의 account 아이콘에 transform의 translate을 이용하여 bounce 효과를 주었습니다. (Figure 14)

SignUpPage에서는 SignUp 버튼의 scale 값에 변화를 주어 강조하였습니다. (Figure 13)

텍스트이(가) 표시된 사진

자동 생성된 설명Figure 12: transform property with hover

**14. Use the @keyframes property and animation-duration, animation-name, animation-iteration-count for Animation.**

위에 설명했다시피, MainPage와 VideoPage의 Account아이콘에 bounce animation을 주었고(Figure 14), SignUpPage의 SignUp 버튼의 강조 animation을 설정하였습니다(Figure 13).

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 13: button animation

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 14: bounce animation