[표지]

**2019 정보통신 2학기 프로젝트**

**작품 설명서**

|  |  |
| --- | --- |
| 작품명 | 선린 수면 도우미 |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| 작 품 명 | 선린 수면 도우미 |
| 조 번호 | 8조 |
| 팀원 1 (팀장) | 10523 정해성 |
| 팀원 2 | 10518 우동관 |

[본문]

**1. 작품 개요**

가. 개발 동기 및 기대효과

선린에는 잠을 제대로 잘 자지 못하고 잠이 부족해서 수업시간에 졸거나 잠을 자는 학생들을 종종 볼 수 있습니다. 그래서 절대적인 수면양을 늘릴 수는 없지만 잠을 잘 때에 규칙적인 수면 패턴을 통해 얼마나 자거나 언제 자면 개운하게 일어날 수 있는지 계산해주는 웹사이트를 만들면 유용할 것 같아서 만들어보게 되었습니다.

선린 학생들이 잠을 자기 전이나 수면 계획을 세울 때 ‘선린 수면 도우미’ 사이트를 참고한다면 좀 더 간편하고 만족스럽게 잘 수 있을 것입니다.

**(*콘텐츠 선정 사유, 개발동기, 작품의 유용성 및 운영방향, 기대효과 등 기재*)**

나. 제작과정

|  |  |  |
| --- | --- | --- |
| **개**  **발**  **일**  **정** | **날짜 또는 기간** | **제작과정** |
| 2019.11.21 | 기본적인 HTML 작성 |
| 2019.11.22 | scene구성과 전환 구현 |
| 2019.11.27 | 계산 알고리즘을 통해 시간 표시 |
| 2019.11.28 | 스크롤 기능 추가 |
| 2019.12.1 | 전체적인 기능 구현 |
|  | 2019.12.2 | 수정 및 기능 추가 |

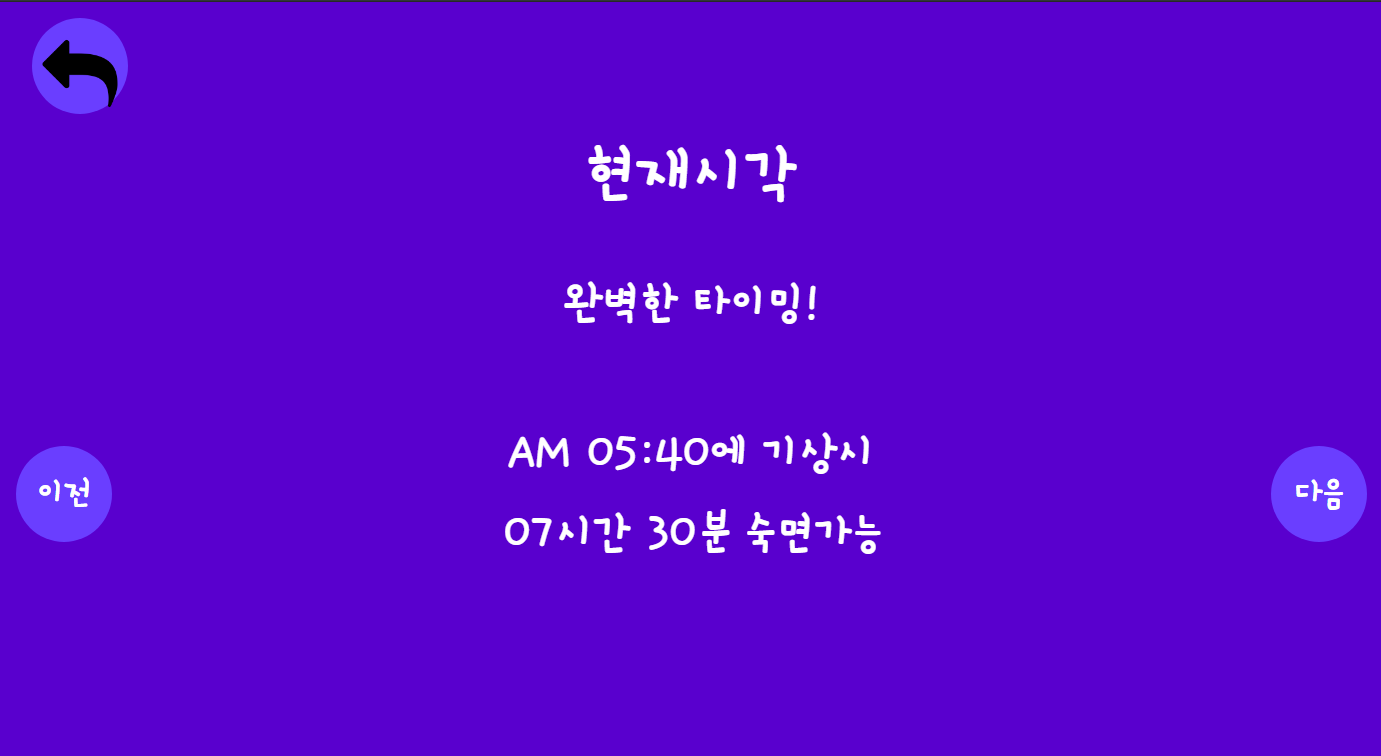
**2. 작품 소개**

가. 사용 설명서

**(각 메뉴 및 기능별 소스 등을 화면 캡쳐, 특징 위주로 자유롭고 충분히 자세하게 설명할 것)**

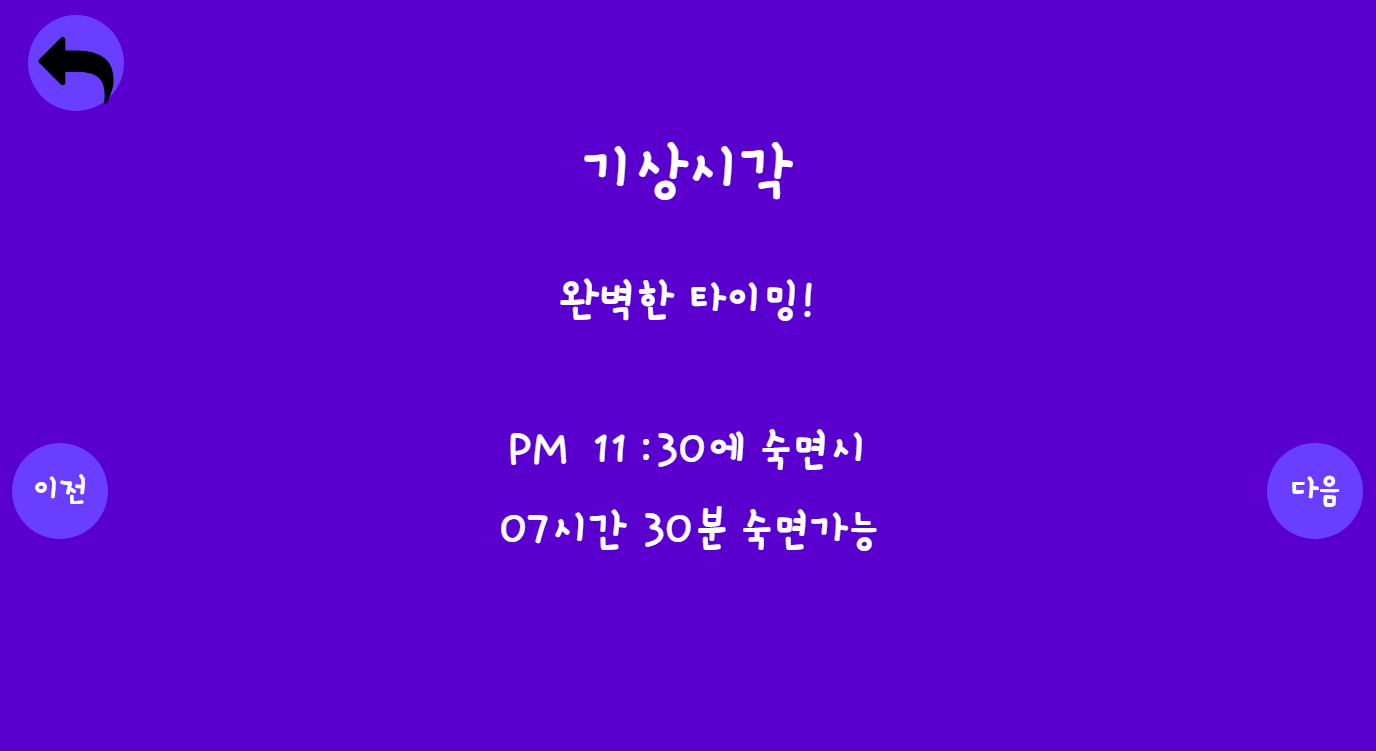
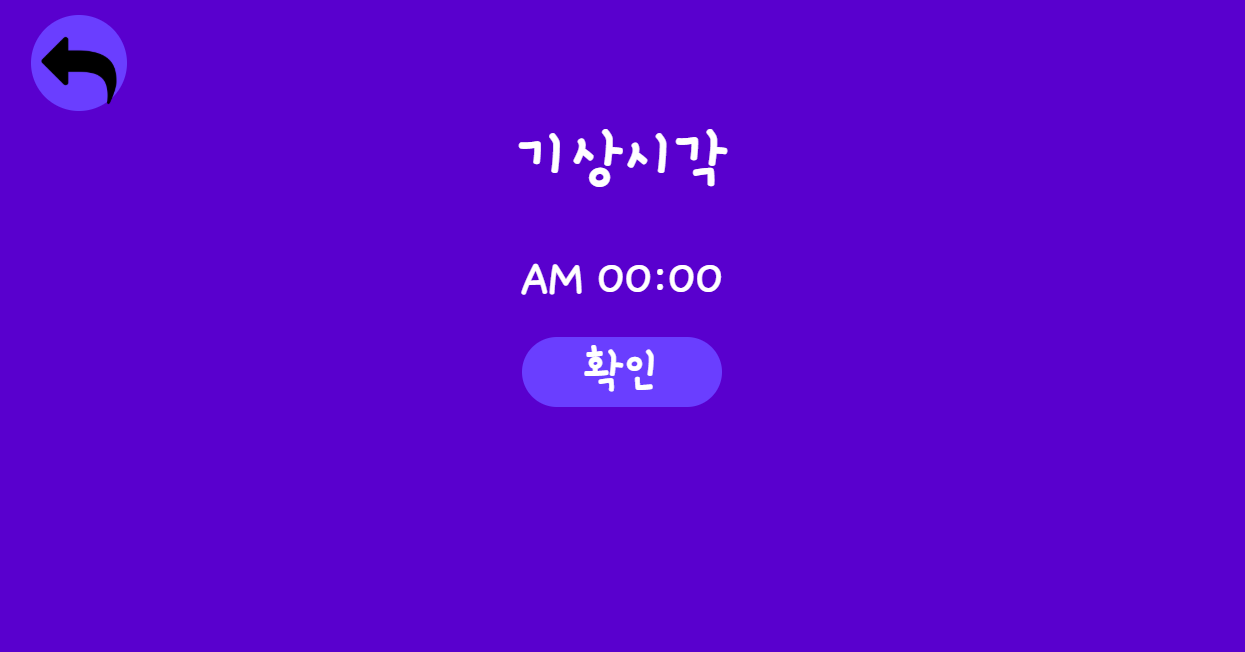
메인 화면에는 버튼이 왼쪽부터 ‘현재시각’, ‘기상시각’, ‘수면시각’. 3개가 있습니다.

1. ‘현재시각’ 버튼



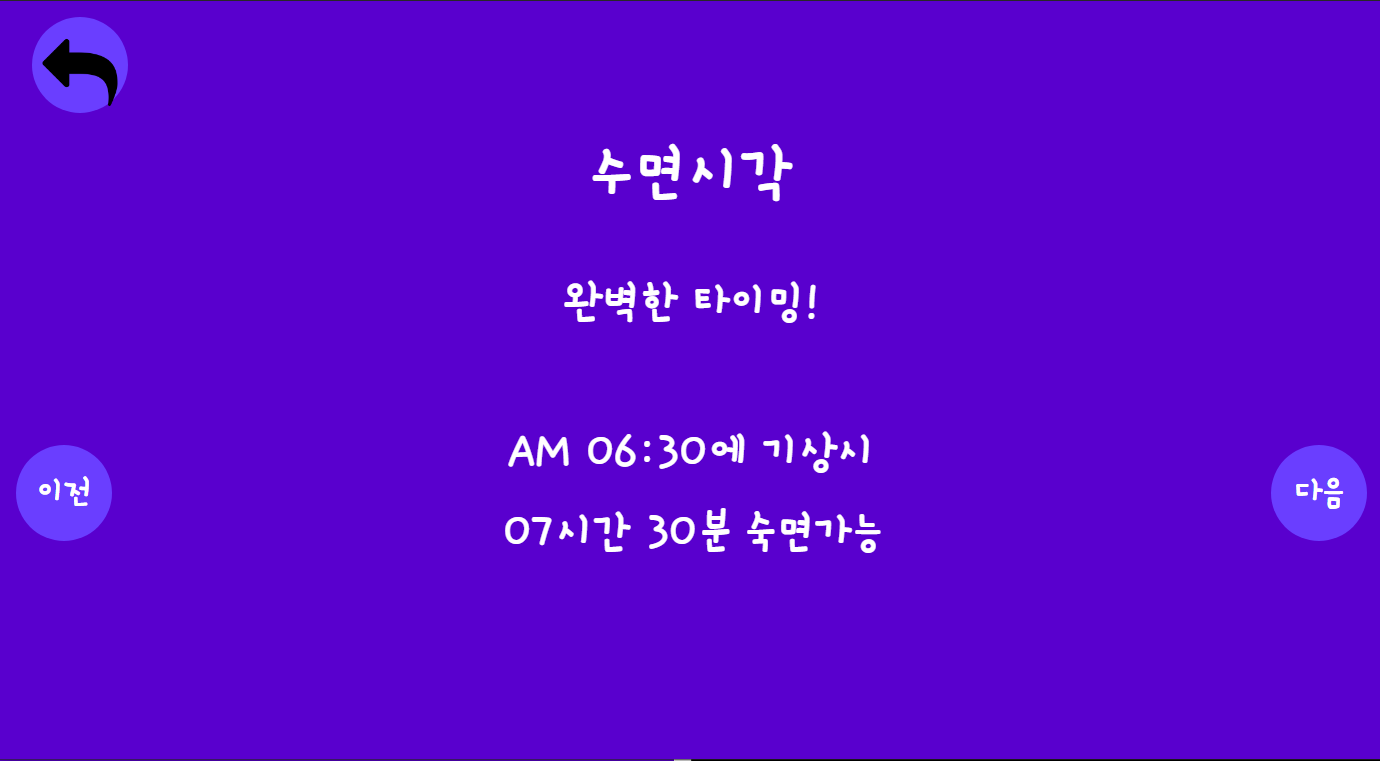
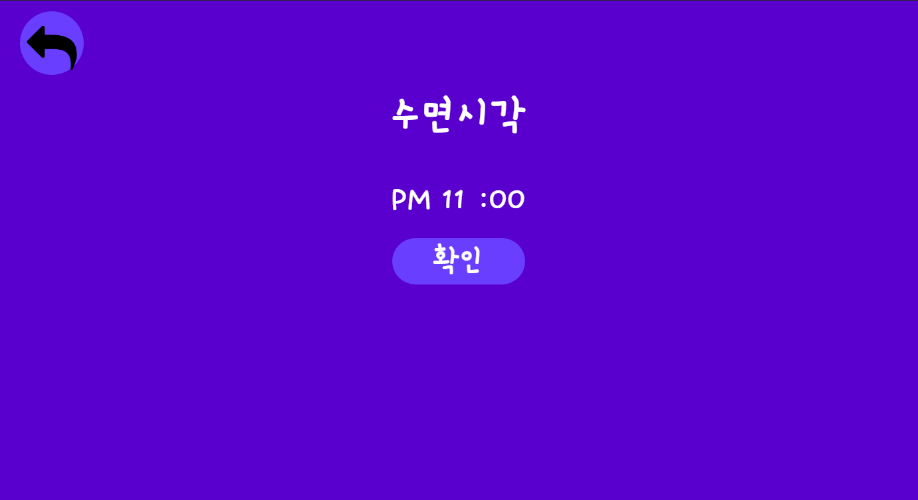
현재시각을 기준으로 일어나면 좋을 시각을 추천해줍니다. 이전 버튼을 누르면 추천하는 기상시각이 현재시각에 가까워지고 다음 버튼을 누르면 현재시각으로부터 멀어진 시각을 표시해줍니다.

1. ‘기상시각’ 버튼



‘AM’, ’00:00’에 마우스 스크롤을 하면 시간을 바꿔 설정할 수 있습니다. 일어나고 싶은 시각을 설정하고 확인 버튼을 누르면 언제 자는 것이 좋을지 추천해주는 화면을 띄웁니다.

1. ‘수면시각’ 버튼



‘수면시각’ 탭은 숫자를 스크롤하여 내가 언제 잘 것인지를 설정해주면 됩니다. 확인 버튼을 누르면 설정해둔 잘 시각에 따라서 일어나면 좋을 시각을 추천해줍니다.

나. 주요 소스코드

@keyframes start\_ani\_btn {

            0% {

                left: -100vw;

            }

            100% {

                left: 0vw;

            }

        }

        @keyframes start\_ani\_title {

            0% {

                margin: 20vw 0;

                font-size: 100px;

            }

            100% {

                margin: 10vw 0 5vw 0;

                font-size: 70px;

            }

        }

애니메이션 적용을 위한 키프레임

.scene {

            width: 100vw;

            height: 100vh;

            position: absolute;

            top: 100vh;

            transition: top 0.5s;

            background-color: #5900CE;

        }

        .scene-isShow {

            top: 0;

        }

        #now-btn {

            position: relative;

            animation: start\_ani\_btn 1s 2.5s forwards;

            left: -100vw;

        }

        #wake-btn {

            position: relative;

            animation: start\_ani\_btn 1s 2s forwards;

            left: -100vw;

        }

        #goal-btn {

            position: relative;

            animation: start\_ani\_btn 1s 1.5s forwards;

            left: -100vw;

        }

키프레임 적용과 해당 신으로 넘어갈 때 애니메이션을 구현하기 위한 CSS코드

function change\_scene(sceneNumber) {

            switch (sceneNumber) {

                case 0:

                    now\_update();

                    document.getElementById("now").classList.add("scene-isShow");

                    break;

                case 1:

                    calculate\_mode = 1;

                    document.getElementById("wake").classList.add("scene-isShow");

                    break;

                case 2:

                    calculate\_mode = 2;

                    document.getElementById("goal").classList.add("scene-isShow");

                    break;

                case 3:

                    if (calculate\_mode == 1) {

                        wake\_result\_update();

                    } else {

                        goal\_result\_update();

                    }

                    document.getElementById("result").classList.add("scene-isShow");

                    break;

            }

        }

        function back\_scene(sceneNumber) {

            switch (sceneNumber) {

                case 0:

                    document.getElementById("now").classList.remove("scene-isShow");

                    break;

                case 1:

                    document.getElementById("wake").classList.remove("scene-isShow");

                    break;

                case 2:

                    document.getElementById("goal").classList.remove("scene-isShow");

                    break;

                case 3:

                    document.getElementById("result").classList.remove("scene-isShow");

                    break;

            }

        }

미리 CSS에 작성해 놓은 클래스를 추가나 제거 해줌으로써 애니메이션 구현

function time(\_isam, \_hour, \_minute) {

            this.isam = \_isam;

            this.hour = \_hour;

            this.minute = \_minute;

            this.tominutes = function () {

                return (this.isam ? 0 : 12 \* 60) + this.hour \* 60 + this.minute;

            };

        }

        function totime(minutes) {

            while (minutes > 24 \* 60) {

                minutes -= 24 \* 60;

            }

            if (minutes < 0) {

                minutes += 24 \* 60;

            }

            var tmp = new time(1, 0, 0);

            if (minutes >= 12 \* 60) {

                minutes -= 12 \* 60;

                tmp.isam = 0;

            }

            tmp.hour = (minutes - minutes % 60) / 60;

            minutes -= tmp.hour \* 60;

            tmp.minute = minutes;

            //console.log(tmp.isam + " " + tmp.hour + " "+ tmp.minute);

            return tmp;

        }

웹 페이지 내에서 사용하기 위한 시간 객체의 생성자 선언 및 객체를 좀더 쉬운 계산을 위해 분단위로 변경하거나 다시 시간 객체로 바꾸어서 반환해주는 함수

var selector\_time = new time(1, 0, 0);

        function selectorWheel(e, th) {

            e.preventDefault();

            var delta = e.wheelDelta / 120;

            if (th.id == "time\_ampm\_wake\_selector" || th.id == "time\_ampm\_goal\_selector") {

                if (delta < 0) {

                    selector\_time.isam = 0;

                } else {

                    selector\_time.isam = 1;

                }

            } else if (th.id == "time\_hour\_wake\_selector" || th.id == "time\_hour\_goal\_selector") {

                if (delta < 0) {

                    if (selector\_time.hour < 12)

                        selector\_time.hour++;

                    else if (selector\_time.isam) {

                        selector\_time.hour = 0;

                        selector\_time.isam = 0;

                    }

                } else {

                    if (selector\_time.hour > 0)

                        selector\_time.hour--;

                    else if (!selector\_time.isam) {

                        selector\_time.isam = 1;

                        selector\_time.hour = 12;

                    }

                }

                if (selector\_time.hour == 12) {

                    selector\_time.minute = 0;

                }

            } else if (th.id == "time\_min\_wake\_selector" || th.id == "time\_min\_goal\_selector") {

                if (delta < 0) {

                    if (selector\_time.minute < 50 && selector\_time.hour < 12) {

                        selector\_time.minute += 10;

                    } else if (selector\_time.hour < 12) {

                        selector\_time.hour++;

                        selector\_time.minute = 0;

                    } else if (selector\_time.isam) {

                        selector\_time.isam = 0;

                        selector\_time.hour = 0;

                        selector\_time.minute = 0;

                    }

                } else {

                    if (selector\_time.minute > 0) {

                        selector\_time.minute -= 10;

                    } else if (selector\_time.hour > 0) {

                        selector\_time.hour--;

                        selector\_time.minute = 50;

                    } else if (!selector\_time.isam) {

                        selector\_time.isam = 1;

                        selector\_time.hour = 12;

                        selector\_time.minute = 0;

                    }

                }

            }

        }

시간을 선택할 때 사용한 스크롤 형식의 시간 선택 방식 구현 코드, 시간 객체를 선언한 뒤 기존에 일어난 이벤트는 무효화시키고 새로운 이벤트 실행

var ampm\_wake\_selector, hour\_wake\_selector, minute\_wake\_selector;

        var ampm\_goal\_selector, hour\_goal\_selector, minute\_goal\_selector;

        var ampm\_selector\_Inter, hour\_selector\_Inter, minute\_selector\_Inter;

        var ampm\_now\_should\_sleep\_selector, hour\_now\_should\_sleep\_selector, min\_now\_should\_sleep\_selector;

        var hour\_now\_can\_sleep\_selector, min\_now\_can\_sleep\_selector;

        var ampm\_now\_sleep\_selector\_Inter, hour\_now\_sleep\_selector\_Inter, min\_now\_sleep\_selector\_Inter;

        var ampm\_result\_should\_sleep\_selector, hour\_result\_should\_sleep\_selector, min\_result\_should\_sleep\_selector;

        var hour\_result\_can\_sleep\_selector, min\_result\_can\_sleep\_selector;

        var ampm\_result\_should\_sleep\_selector, hour\_result\_sleep\_selector\_Inter, min\_result\_sleep\_selector\_Inter;

        function setting() {

            ampm\_wake\_selector = document.getElementById("time\_ampm\_wake\_selector");

            hour\_wake\_selector = document.getElementById("time\_hour\_wake\_selector");

            minute\_wake\_selector = document.getElementById("time\_min\_wake\_selector");

            ampm\_goal\_selector = document.getElementById("time\_ampm\_goal\_selector");

            hour\_goal\_selector = document.getElementById("time\_hour\_goal\_selector");

            minute\_goal\_selector = document.getElementById("time\_min\_goal\_selector");

            ampm\_now\_should\_sleep\_selector = document.getElementById("time\_ampm\_now\_should\_sleep\_selector");

            hour\_now\_should\_sleep\_selector = document.getElementById("time\_hour\_now\_should\_sleep\_selector");

            min\_now\_should\_sleep\_selector = document.getElementById("time\_min\_now\_should\_sleep\_selector");

            hour\_now\_can\_sleep\_selector = document.getElementById("time\_hour\_now\_can\_sleep\_selector");

            min\_now\_can\_sleep\_selector = document.getElementById("time\_min\_now\_can\_sleep\_selector");

            ampm\_result\_should\_sleep\_selector = document.getElementById("time\_ampm\_result\_should\_sleep\_selector");

            hour\_result\_should\_sleep\_selector = document.getElementById("time\_hour\_result\_should\_sleep\_selector");

            min\_result\_should\_sleep\_selector = document.getElementById("time\_min\_result\_should\_sleep\_selector");

            hour\_result\_can\_sleep\_selector = document.getElementById("time\_hour\_result\_can\_sleep\_selector");

            min\_result\_can\_sleep\_selector = document.getElementById("time\_min\_result\_can\_sleep\_selector");

            ampm\_selector\_Inter = setInterval(function () {

                if (selector\_time.isam) {

                    ampm\_wake\_selector.scrollBy(0, -1);

                    ampm\_goal\_selector.scrollBy(0, -1);

                } else {

                    ampm\_wake\_selector.scrollBy(0, 1);

                    ampm\_goal\_selector.scrollBy(0, 1);

                }

            });

            hour\_selector\_Inter = setInterval(function () {

                if (selector\_time.hour > hour\_wake\_selector.scrollTop / 50) {

                    hour\_wake\_selector.scrollBy(0, selector\_time.hour - Math.floor(hour\_wake\_selector

                        .scrollTop / 50));

                    hour\_goal\_selector.scrollBy(0, selector\_time.hour - Math.floor(hour\_goal\_selector

                        .scrollTop / 50));

                } else {

                    hour\_wake\_selector.scrollBy(0, selector\_time.hour - Math.ceil(hour\_wake\_selector.scrollTop /

                        50));

                    hour\_goal\_selector.scrollBy(0, selector\_time.hour - Math.ceil(hour\_goal\_selector.scrollTop /

                        50));

                }

            });

            minute\_selector\_Inter = setInterval(function () {

                if (selector\_time.minute / 10 > minute\_wake\_selector.scrollTop / 50) {

                    minute\_wake\_selector.scrollBy(0, selector\_time.minute / 10 - Math.floor(minute\_wake\_selector

                        .scrollTop / 50));

                    minute\_goal\_selector.scrollBy(0, selector\_time.minute / 10 - Math.floor(minute\_goal\_selector

                        .scrollTop / 50));

                } else {

                    minute\_wake\_selector.scrollBy(0, selector\_time.minute / 10 - Math.ceil(minute\_wake\_selector

                        .scrollTop / 50));

                    minute\_goal\_selector.scrollBy(0, selector\_time.minute / 10 - Math.ceil(minute\_goal\_selector

                        .scrollTop / 50));

                }

            });

            ampm\_now\_sleep\_selector\_Inter = setInterval(function () {

                if (now\_calculate\_result.isam) {

                    ampm\_now\_should\_sleep\_selector.scrollBy(0, -1);

                } else {

                    ampm\_now\_should\_sleep\_selector.scrollBy(0, 1);

                }

            });

            hour\_now\_sleep\_selector\_Inter = setInterval(function () {

                if (now\_calculate\_result.hour > hour\_now\_should\_sleep\_selector.scrollTop / 50) {

                    hour\_now\_should\_sleep\_selector.scrollBy(0, now\_calculate\_result.hour - Math.floor(

                        hour\_now\_should\_sleep\_selector.scrollTop / 50));

                } else {

                    hour\_now\_should\_sleep\_selector.scrollBy(0, now\_calculate\_result.hour - Math.ceil(

                        hour\_now\_should\_sleep\_selector.scrollTop / 50));

                }

                if (now\_calculate\_time.hour > hour\_now\_can\_sleep\_selector.scrollTop / 50) {

                    hour\_now\_can\_sleep\_selector.scrollBy(0, now\_calculate\_time.hour - Math.floor(

                        hour\_now\_can\_sleep\_selector.scrollTop / 50));

                } else {

                    hour\_now\_can\_sleep\_selector.scrollBy(0, now\_calculate\_time.hour - Math.ceil(

                        hour\_now\_can\_sleep\_selector.scrollTop / 50));

                }

            });

            min\_now\_sleep\_selector\_Inter = setInterval(function () {

                if (Math.floor(now\_calculate\_result.minute / 10) > min\_now\_should\_sleep\_selector.scrollTop /

                    50) {

                    min\_now\_should\_sleep\_selector.scrollBy(0, now\_calculate\_result.minute / 10 - Math.floor(

                        min\_now\_should\_sleep\_selector.scrollTop / 50));

                } else {

                    min\_now\_should\_sleep\_selector.scrollBy(0, now\_calculate\_result.minute / 10 - Math.ceil(

                        min\_now\_should\_sleep\_selector.scrollTop / 50));

                }

                if (now\_calculate\_time.minute / 10 > min\_now\_can\_sleep\_selector.scrollTop / 50) {

                    min\_now\_can\_sleep\_selector.scrollBy(0, now\_calculate\_time.minute / 10 - Math.floor(

                        min\_now\_can\_sleep\_selector.scrollTop / 50));

                } else {

                    min\_now\_can\_sleep\_selector.scrollBy(0, now\_calculate\_time.minute / 10 - Math.ceil(

                        min\_now\_can\_sleep\_selector.scrollTop / 50));

                }

            });

            ampm\_result\_sleep\_selector\_Inter = setInterval(function () {

                if (result\_calculate\_result.isam) {

                    ampm\_result\_should\_sleep\_selector.scrollBy(0, -1);

                } else {

                    ampm\_result\_should\_sleep\_selector.scrollBy(0, 1);

                }

            });

            hour\_result\_sleep\_selector\_Inter = setInterval(function () {

                if (result\_calculate\_result.hour > hour\_result\_should\_sleep\_selector.scrollTop / 50) {

                    hour\_result\_should\_sleep\_selector.scrollBy(0, result\_calculate\_result.hour - Math.floor(

                        hour\_result\_should\_sleep\_selector.scrollTop / 50));

                } else {

                    hour\_result\_should\_sleep\_selector.scrollBy(0, result\_calculate\_result.hour - Math.ceil(

                        hour\_result\_should\_sleep\_selector.scrollTop / 50));

                }

                if (result\_calculate\_time.hour > hour\_result\_can\_sleep\_selector.scrollTop / 50) {

                    hour\_result\_can\_sleep\_selector.scrollBy(0, result\_calculate\_time.hour - Math.floor(

                        hour\_result\_can\_sleep\_selector.scrollTop / 50));

                } else {

                    hour\_result\_can\_sleep\_selector.scrollBy(0, result\_calculate\_time.hour - Math.ceil(

                        hour\_result\_can\_sleep\_selector.scrollTop / 50));

                }

            });

            min\_result\_sleep\_selector\_Inter = setInterval(function () {

                if (Math.floor(result\_calculate\_result.minute / 10) > min\_result\_should\_sleep\_selector

                    .scrollTop / 50) {

                    min\_result\_should\_sleep\_selector.scrollBy(0, result\_calculate\_result.minute / 10 - Math

                        .floor(

                            min\_result\_should\_sleep\_selector.scrollTop / 50));

                } else {

                    min\_result\_should\_sleep\_selector.scrollBy(0, result\_calculate\_result.minute / 10 - Math

                        .ceil(

                            min\_result\_should\_sleep\_selector.scrollTop / 50));

                }

                if (result\_calculate\_time.minute / 10 > min\_result\_can\_sleep\_selector.scrollTop / 50) {

                    min\_result\_can\_sleep\_selector.scrollBy(0, result\_calculate\_time.minute / 10 - Math.floor(

                        min\_result\_can\_sleep\_selector.scrollTop / 50));

                } else {

                    min\_result\_can\_sleep\_selector.scrollBy(0, result\_calculate\_time.minute / 10 - Math.ceil(

                        min\_result\_can\_sleep\_selector.scrollTop / 50));

                }

            });

        }

선택을 시간하거나 계산 결과값을 알려줄 때 애니메이션을 주기위한 관련된 DOM객체 불러오기와 인터벌 선언 및 인터벌 안에 적절히 스크롤 애니메이션을 구현하기 위한 코드들

**(주요기능 2-3가지 정도에 대해 자세히 설명)**

다. 향후 작품 개선/발전 계획

- 맨 처음 개발했을 때 보다는 많이 가독성이 좋아지긴 했지만 아직은 처음 사용하는 사람이 직관적으로 사용하기에는 조금 헷갈릴 수도 있겠다는 생각이 들었습니다. 그래서 좀 더 직관적으로 사용할 수 있도록 UI 디자인을 수정하고 웹사이트가 계산해서 알려준 시각을 한시계의 시침과 분침을 이용해서 한눈에 보기 편하게 한다면 지금보다 개선될 것 같습니다.

**3. 팀별 개인 역할**

|  |  |
| --- | --- |
| **팀원** | **역할** |
| **정해성 :** | 전체적인 HTML 구성과 약간의 CSS 수정 및 전체적인 보고서 작성 |
| **우동관 :** | XD로 기본 틀 구상, 이미지 제작, JS 소스 작성 및 약간의 CSS 수정 |

**3. 느낀 점**

**팀원 1(정해성):**

사실 난 예전부터 웹이나 서버 같은 것에 관심을 두지 않았었다. 이번 프로젝트에서도 기여한 것이라고는 보고서 작성과 HTML, CSS 일부 작성 뿐이었지만 옆에서 JS로 화면 이동 등 애니메이션을 만들고 구현하는 것을 보고 신기하고 흥미가 생겼다. 아직도 HTML, CSS에는 관심이 없긴 하지만 이번 계기로 JS에 대한 학구열을 느낄 수 있었던 것 같다.

**팀원 2(우동관):**

예전부터 친해지고 싶은 친구가 웹을 전공하고 있었기에 웹에는 관심이 꽤나 있었다. 하지만 CSS작성에는 크게 적성에 맞지 않다고 느끼고 있었지만, 이 기회를 통해 CSS작성을 연습하고, 선배님들께 모르는 것은 물어보면서 CSS작성 실력을 꽤나 크게 늘렸다. 또한 JS를 주로 다루면서 지금까지 학습해왔던 내용을 복습할 수 있었으며, 최근에 배운 Scroll 관련 기능들을 응용하면서 이번에 제작한 웹페이지의 완성도를 크게 올릴 수 있어서 꽤나 뿌듯하였고, 앞으로도 이런 프로젝트들을 통해서 향상된 실력을 바탕으로 혼자서 제작해 보고 싶은 웹페이지가 있다면 제작해 보고싶다.