

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



Báo cáo bài tập lớn
Hệ thống thông tin trên Web
Đề tài: “Calendar”

GVHD: TS.Phạm Huy Hoàng

Nhóm: 13

Thành viên:

Họ và tên	Mã số sinh viên
Phạm Hữu An	20176676
Bùi Thị Phụng	20176849

Nội dung

I.	Tổng quan đề tài	2
1.	Hiện trạng	2
2.	Đặt vấn đề	2
3.	Công nghệ sử dụng	3
II.	Thiết kế phần mềm	3
1.	Chức năng xem lịch	3
2.	Chức năng quản lý ghi chú	3
3.	Chức năng quản lý sự kiện	5
III.	Demo sản phẩm	7
IV.	Mã nguồn	10

I. Tổng quan đề tài

1. Hiện trạng

Cuộc sống càng phát triển càng đặt ra nhiều yêu cầu mới cho mỗi cá nhân. Để hoàn thiện chính mình và đáp ứng yêu cầu của thời đại, mỗi cá nhân đều phải trang bị cho mình những kỹ năng cần thiết, trong đó có kỹ năng quản lý thời gian. Nhưng theo nghiên cứu của Viện Khoa học Giáo dục Việt Nam, có tới 83% sinh viên khi ra trường bị nhà tuyển dụng đánh giá thiếu kỹ năng sử dụng thời gian hợp lý, hiệu suất lao động thấp, không ít bạn trả mất nhiều cơ hội trên bước đường lập nghiệp. Sinh viên phải giải quyết các vấn đề của mình một cách độc lập: học tập, sinh hoạt, việc làm thêm... Việc phân bổ thời gian càng trở nên quan trọng và cấp thiết.

Nhiều bạn dành ra quá nhiều thời gian để chơi game, đi làm thêm hay đi chơi với bạn bè... Có rất nhiều việc quan trọng mà các bạn đã bỏ qua. Như là quên mất ngày mai kiểm tra giữa kì, tuần sau nộp báo cáo project.. Dù việc gì thì nó cũng dẫn đến hậu quả khác nhau mà không ai mong muốn.

2.Đặt vấn đề

Với sự phát triển của khoa học, kỹ thuật thì nhiều ứng dụng khác nhau đã ra đời. Mỗi ứng dụng có một mục đích khác nhau nhưng cuối cùng vẫn là giúp người dùng nắm bắt được khối lượng công việc của mình để quản lý thời gian hiệu quả hơn.

Nhận thức được vấn đề cần thiết đó, nhóm 13 chúng em quyết định xây dựng calendar để mọi người có thể không chỉ dễ dàng xem ngày, tháng, năm mà còn có thể thêm các công việc cần thiết trong

từng ngày. Vào những ngày quan trọng chúng ta có thể để nó lặp lại theo từng tháng từng năm

3.Mục đích

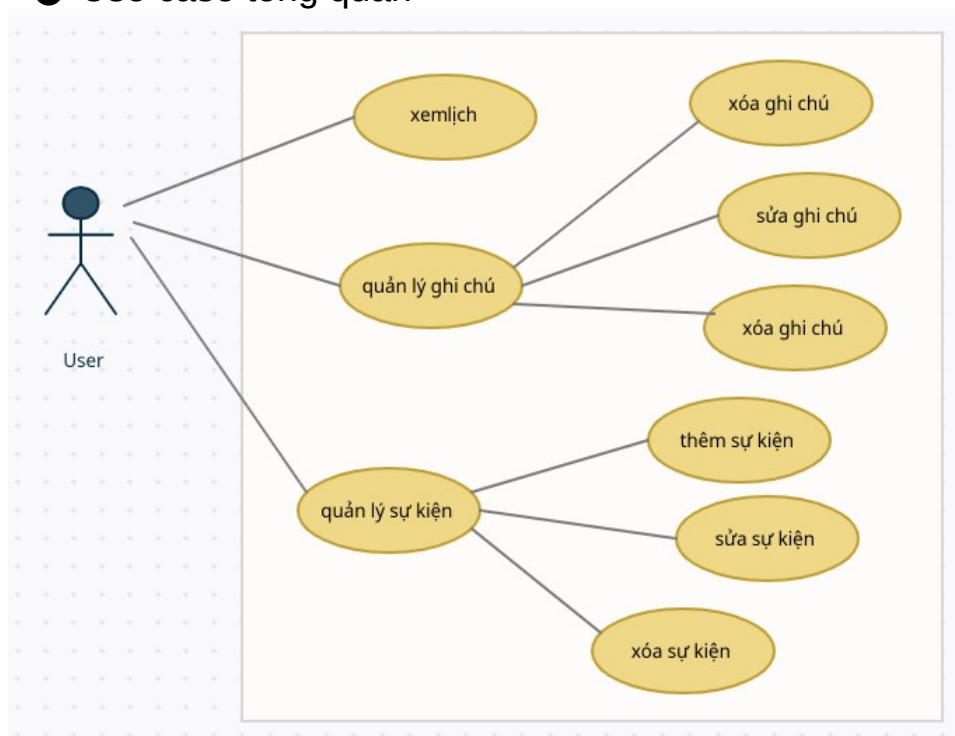
Calendar ra đời với mục đích giúp cho tất cả mọi người đặc biệt là các bạn sinh viên có thể quản lý sử dụng thời gian một cách logic hiệu quả hơn thông qua việc liệt kê các công việc hàng ngày, hàng tuần, hàng năm phải làm trong học tập và đời sống thường nhật.

4.Công nghệ sử dụng

- Front-End: html, css, javascript
- Database: LocalStorage

II. Thiết kế phần mềm

● Use case tổng quan



1. Chức năng xem lịch

- Bên phía user có thể xem ngày trong tháng trong các năm và xem lịch âm.
- Khi click vào ngày nào thì sẽ hiện ra thứ, ngày, tháng của hôm đó

2. Chức năng quản lý ghi chú

2.1. Chức năng thêm ghi chú

User có thể thêm các ghi chú trong ngày bất kì. Sau khi nhập ghi chú, lắng nghe sự kiện user click nút "Add". Lúc này dữ liệu

tiến hành lưu vào LocalStorage thông tin vừa add lên mục note của người dùng.

Khi ngày nào đó có sự kiện thì trong ô chứa ngày đó sẽ đánh dấu là có sự kiện

```
const editNote = (index) => {
  let noteList2 = JSON.parse(noteList);
  textAreaValue = noteList2[selectedDate.toDateString()][index];
  isAdd = false;
  selectedIndex = index;
  renderNoteView();
};
```

2.2. Chức năng sửa ghi chú

User tiến hành sửa ghi chú.

- Lắng nghe sự kiện user click nút "Edit". Lúc này dữ liệu được lấy ra từ LocalStorage thông tin note để user chỉnh sửa lại.

```
const editNote = (index) => {
  let noteList2 = JSON.parse(noteList);
  textAreaValue = noteList2[selectedDate.toDateString()][index];
  isAdd = false;
  selectedIndex = index;
  renderNoteView();
};
```

- Sau khi sửa xong lắng nghe sự kiện user click nút "Save". Lúc này dữ liệu sẽ tiến hành lưu vào LocalStorage tại vị trí lúc edit.

```
const saveNote = () => {
  const noteArea = document.getElementById('addNoteArea');
  if (noteArea.value.trim() !== '') {
    let noteList2 = JSON.parse(noteList);
    noteList2[selectedDate.toDateString()][selectedIndex] = noteArea.value.trim();
    noteList = JSON.stringify(noteList2);
    isAdd = true;
    noteArea.value = '';
    textAreaValue = '';
    try {
      localStorage.setItem('noteList', noteList);
    } catch (error) {}
    renderNoteView();
  }
};
```

- Hoặc sự kiện user click nút "cancel" để hủy việc edit đang làm

```
const cancelEdit = () => {
  const noteArea = document.getElementById('addNoteArea');
  isAdd = true;
  noteArea.value = '';
  textAreaValue = '';
  renderNoteView();
}
```

2.3. Chức năng xóa ghi chú

User tiến hành xóa ghi chú.

Ứng dụng lắng nghe sự kiện người dùng click nút “Delete”. Lúc này dữ liệu lấy ra từ LocalStorage rồi thực hiện việc xóa ghi chú.

```
const deleteNote = (index) => {
  let noteList2 = JSON.parse(noteList);
  if (noteList2[selectedDate.toDateString()].length === 1) {
    delete noteList2[selectedDate.toDateString()];
  } else {
    noteList2[selectedDate.toDateString()].splice(index, 1);
  }
  noteList = JSON.stringify(noteList2);
  isAdd = true;
  try {
    localStorage.setItem('noteList', noteList);
  } catch (error) {}
  renderNoteView();
  renderCalendar();
};
```

3. Chức năng quản lý sự kiện

3.1. Chức năng thêm sự kiện

User có thể thêm các sự kiện trong ngày đó theo no repeat, every week, every month, every year

Khi user click nút “Add”, lúc này dữ liệu tiến hành lưu vào LocalStorage thông tin vừa add lên mục event của user theo thứ tự add từ every year, every month, every week đến no repeat.

Khi có sự kiện thì ô chứa ngày đó sẽ có đánh dấu là có sự kiện gì đó trong ngày đấy.

```
const addEvent = (notRender) => {
  const eventArea = document.getElementById('addNoteArea');
  if (eventArea.value.trim() !== '') {
    let eventList2 = JSON.parse(eventList);
    if (selectedRadio === 0) {
      if (eventList2[selectedDate.getDate()]) {
        if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()]) {
          if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()].push(eventArea.value.trim());
        } else {
          eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()] = [eventArea.value.trim()];
        }
      } else {
        eventList2[selectedDate.getDate()][selectedDate.getMonth()] = {};
        eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()] = [eventArea.value.trim()];
      }
    } else if (selectedRadio === 1) {
      if (eventList2[selectedDate.getDate()]) {
        if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()].push(eventArea.value.trim());
      } else {
        eventList2[selectedDate.getDate()] = [eventArea.value.trim()];
      }
    } else if (selectedRadio === 2) {
      if (eventList2[selectedDate.getDate()]) {
        if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()].push(eventArea.value.trim());
      } else {
        eventList2[selectedDate.getDate()] = [eventArea.value.trim()];
      }
    } else {
      eventList2[selectedDate.getDate()] = [eventArea.value.trim()];
    }
  }
}
```

```

    } else if (selectedRadio === 3) {
      if (eventList2[selectedDate.getDate()] {
        if (eventList2[selectedDate.getDate()][selectedDate.getMonth()] {
          if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1]) {
            eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1].push(eventArea.value.trim());
          } else {
            eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1] = [eventArea.value.trim()];
          }
        } else {
          eventList2[selectedDate.getDate()][selectedDate.getMonth()] = {};
          eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1] = [eventArea.value.trim()];
        }
      } else {
        eventList2[selectedDate.getDate()] = {};
        eventList2[selectedDate.getDate()][selectedDate.getMonth()] = {};
        eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1] = [eventArea.value.trim()];
      }
    }
    eventList = JSON.stringify(eventList2);
    eventArea.value = '';
    try {
      localStorage.setItem('eventList', eventList);
    } catch (error) {}
    if (notRender === true) {
    } else {
      renderNoteView();
      renderCalendar();
    }
  }
}

```

3.2. Chức năng sửa sự kiện

User tiến hành sửa thông tin của sự kiện. Khi user click nút "Edit", lúc này dữ liệu được lấy ra từ LocalStorage thông tin edit đó check xem thuộc loại nào trong 4 loại very year, every month, every week, no repeat.

```

const editEvent = (index, type) => {
  let eventList2 = JSON.parse(eventList);
  if (type === 0) {
    textAreaValue = eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()][index];
  } else if (type === 1) {
    textAreaValue = eventList2['D' + selectedDate.getDay()][index];
  } else if (type === 2) {
    textAreaValue = eventList2[selectedDate.getDate()][-1][index];
  } else if (type === 3) {
    textAreaValue = eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1][index];
  }
  selectedIndex = index;
  selectedRadio = type;
  selectedRadio2 = type;
  isAdd = false;
  renderNoteView();
  renderCalendar();
}

```

Sự kiện khi user click nút “Save”, lúc này dữ liệu sẽ tiến hành lưu vào LocalStorage tại vị trí lúc edit

```

const saveEvent = () => {
  const eventArea = document.getElementById('addNoteArea');
  if (eventArea.value.trim() !== '') {
    let eventList2 = JSON.parse(eventList);
    if (selectedRadio === selectedRadio2) {
      if (selectedRadio === 0) {
        eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()][selectedIndex] = eventArea.value.trim();
      } else if (selectedRadio === 1) {
        eventList2['D' + selectedDate.getDay()][selectedIndex] = eventArea.value.trim();
      } else if (selectedRadio === 2) {
        eventList2[selectedDate.getDate()][-1][selectedIndex] = eventArea.value.trim();
      } else if (selectedRadio === 3) {
        eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1][selectedIndex] = eventArea.value.trim();
      }
    }
    eventList = JSON.stringify(eventList2);
  } else {
    eventList = JSON.stringify(eventList2);
    deleteEvent(selectedIndex, selectedRadio2, true);
    addEvent(true);
  }
  eventArea.value = '';
  textAreaValue = '';
  isAdd = true;
  try {
    localStorage.setItem('eventList', eventList);
  } catch (error) {}
  renderNoteView();
}

```

3.3. Chức năng xóa sự kiện

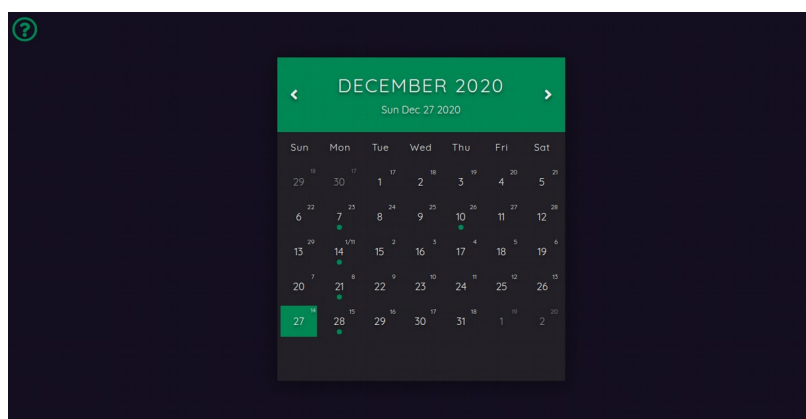
Khi user click nút “Delete”, lúc này dữ liệu lấy ra từ LocalStorage rồi thực hiện việc xóa.

```

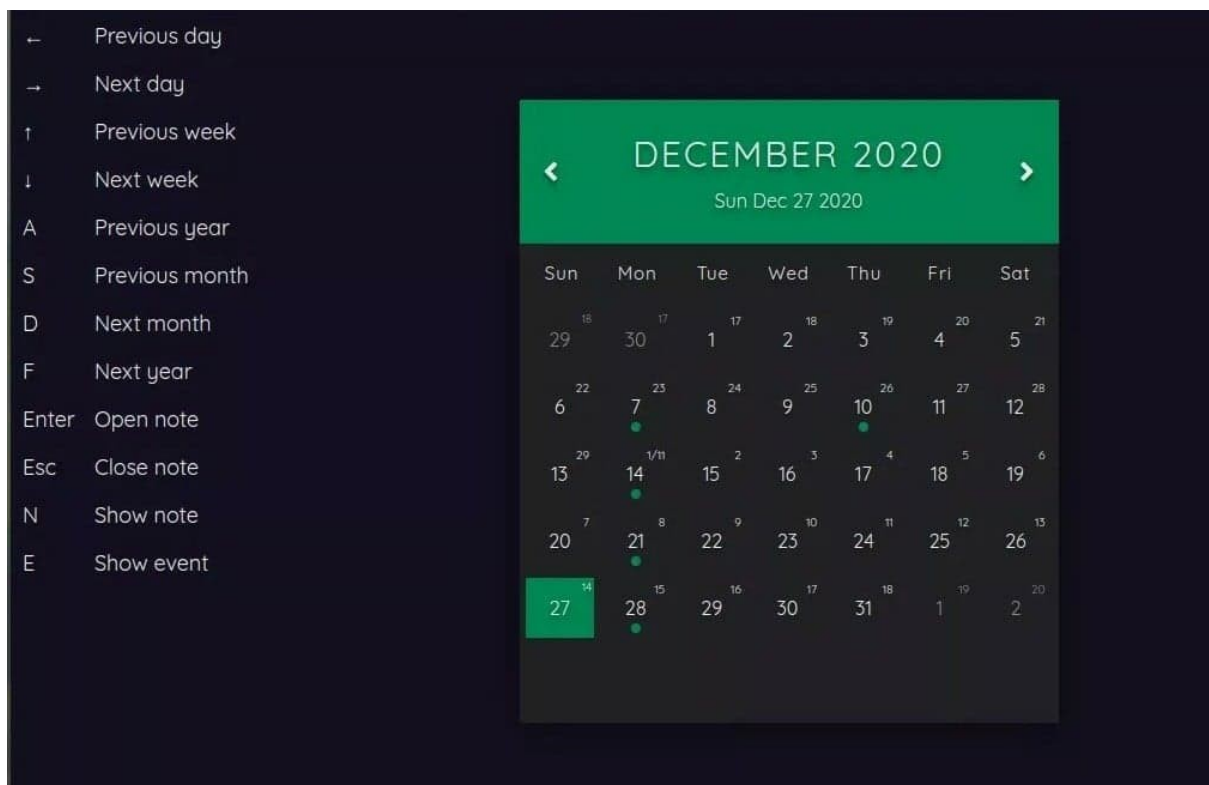
const deleteEvent = (index, type, notRender) => {
  let eventList2 = JSON.parse(eventList);
  if (type === 0) {
    if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()].length === 1) {
      delete eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()];
      if (JSON.stringify(eventList2[selectedDate.getDate()][selectedDate.getMonth()]) === '{}') {
        delete eventList2[selectedDate.getDate()][selectedDate.getMonth()];
        if (JSON.stringify(eventList2[selectedDate.getDate()]) === '{}') {
          delete eventList2[selectedDate.getDate()];
        }
      }
    } else {
      eventList2[selectedDate.getDate()][selectedDate.getMonth()][selectedDate.getFullYear()].splice(index, 1);
    }
  } else if (type === 1) {
    if (eventList2['D' + selectedDate.getDay()].length === 1) {
      delete eventList2['D' + selectedDate.getDay()];
    } else {
      eventList2['D' + selectedDate.getDay()].splice(index, 1);
    }
  } else if (type === 2) {
    if (eventList2[selectedDate.getDate()][-1].length === 1) {
      delete eventList2[selectedDate.getDate()][-1];
      if (JSON.stringify(eventList2[selectedDate.getDate()]) === '{}') {
        delete eventList2[selectedDate.getDate()];
      }
    } else {
      eventList2[selectedDate.getDate()][-1].splice(index, 1);
    }
  } else if (type === 3) {
    if (eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1].length === 1) {
      delete eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1];
      if (JSON.stringify(eventList2[selectedDate.getDate()][selectedDate.getMonth()]) === '{}') {
        delete eventList2[selectedDate.getDate()][selectedDate.getMonth()];
        if (JSON.stringify(eventList2[selectedDate.getDate()]) === '{}') {
          delete eventList2[selectedDate.getDate()];
        }
      }
    } else {
      eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1].splice(index, 1);
    }
  }
  } else {
    eventList2[selectedDate.getDate()][selectedDate.getMonth()][-1].splice(index, 1);
  }
}
}
isAdd = true;
eventList = JSON.stringify(eventList2);
try {
  localStorage.setItem('eventList', eventList);
} catch (error) {}
if (notRender === true) {
} else {
  renderNoteView();
  renderCalendar();
}
}

```

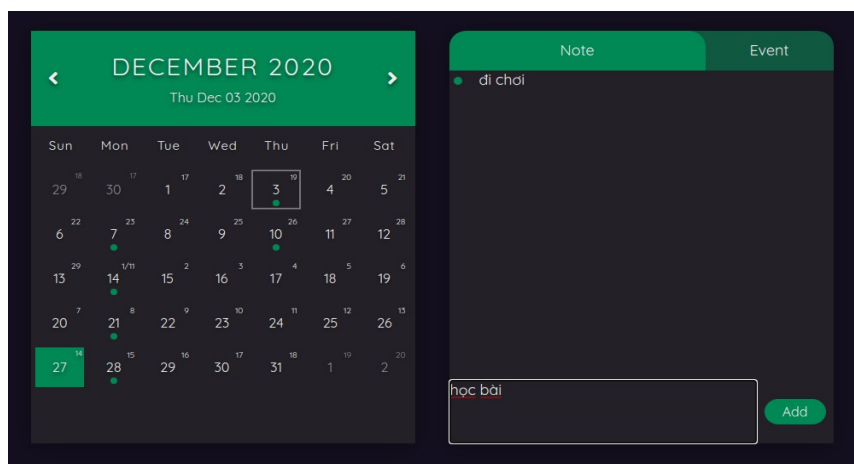
III. Demo sản phẩm



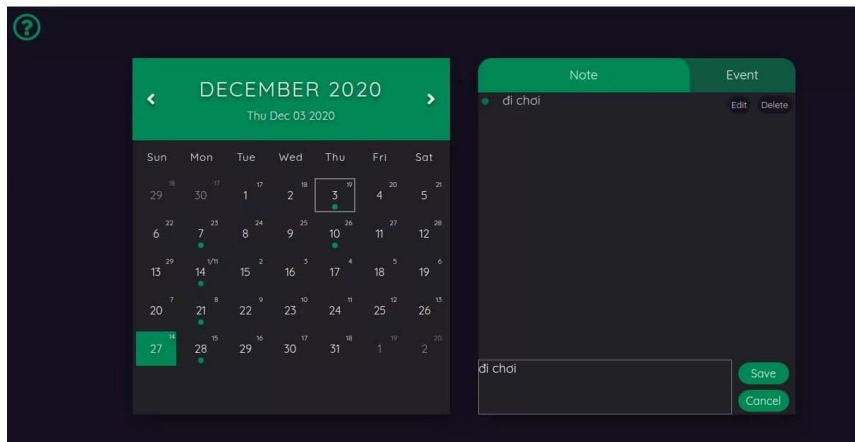
Hình 1: Giao diện xem ngày trong tháng theo lịch dương, lịch âm



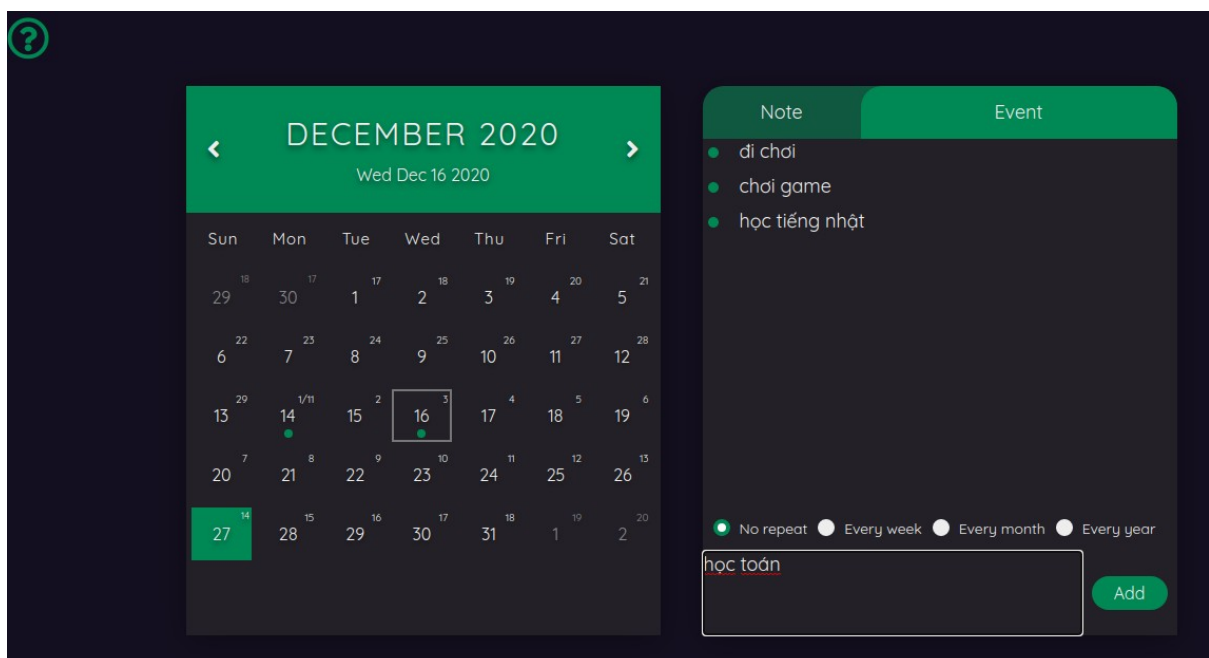
Hình 2: Giao diện hướng dẫn sử dụng calendar



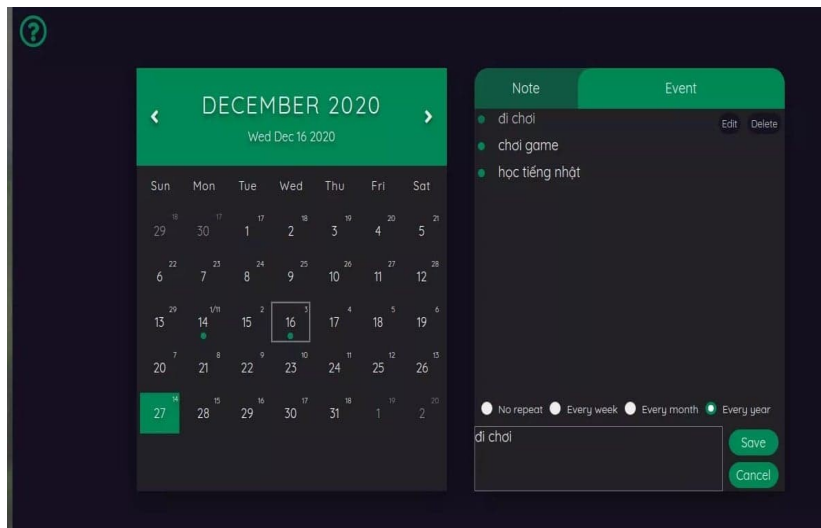
Hình 3: Giao diện add note



Hình 4: Giao diện edit note và delete note



Hình 5: Giao diện add event



Hình 6: Giao diện add Event và delete Event

III. Mã nguồn

Link github: <https://github.com/phamhuuan/MasterCalendarWeb>