

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Đề tài:

Xây dựng chương trình Client/Server và thực hiện in lịch thể kỷ

SINH VIÊN THỰC HIỆN:

Nguyễn Hữu Tùng

LỚP: 21T_DT

Nguyễn Ngọc Tiệp

LỚP: 21T_DT

GIẢNG VIÊN HƯỚNG DẪN:

ThS. Nguyễn Văn Nguyên

MỤC LỤC

MỤC LỤC	2
NHẬN XÉT CỦA GIẢNG VIÊN	4
DANH SÁCH HÌNH VẼ.....	5
DANH SÁCH CÁC TỪ VIẾT TẮT	6
GIỚI THIỆU ĐỀ TÀI.....	7
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	8
1.1 Yêu cầu đề tài	8
1.2 Giới thiệu mô hình Client/Server.....	8
1.2.1 Cấu trúc mô hình Client/Server	8
1.2.1 Máy chủ	8
1.2.2 Máy khách.....	8
1.2.3 Quy trình hoạt động	8
1.3 Giới thiệu giao thức TCP/IP	9
1.3.1 Giới thiệu về TCP	9
1.3.2 Giới thiệu về IP	9
1.3.3 Giới thiệu về cổng.....	9
1.4 Giới thiệu giao thức HTTP/HTTPS	10
CHƯƠNG 2. PHÂN TÍCH BÀI TOÁN	11
2.1 Phân tích chức năng của chương trình.....	11
2.2 Máy chủ	11
2.3 Máy khách.....	11
2.4 Thiết kế chương trình.....	12
2.5 Phương pháp xử lý.....	12
2.5.1 Máy khách.....	12
2.5.2 Máy chủ	19
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	27
3.1 Các công cụ sử dụng.....	27
3.2 Môi trường	27
3.3 Kết quả	27

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	30
TÀI LIỆU THAM KHẢO	31
PHỤ LỤC	32

NHẬN XÉT CỦA GIẢNG VIÊN

DANH SÁCH HÌNH VẼ

Hình 1. Quy trình hoạt động Client/Server.....	9
Hình 2. Nhập dữ liệu.....	13
Hình 3. Lịch tháng.....	18
Hình 4. Lịch năm trên.....	18
Hình 5. Lịch năm dưới.....	19
Hình 6. Html trả về.....	21
Hình 7. Giao diện chính chương trình.....	29
Hình 8. Giao diện xem lịch tháng.....	30
Hình 9. Giao diện xem lịch năm.....	30

DANH SÁCH CÁC TỪ VIẾT TẮT

Từ Viết Tắt	Ý Nghĩa
Server	Máy chủ
Client	Máy khách
TCP/IP	Giao thức điều khiển truyền vận
UDP	Giao thức dữ liệu người dùng
PBL	Dự án PBL 4
HTTP/HTTPS	Giao thức truyền tải siêu văn bản an toàn
GUI	Giao diện người dùng

GIỚI THIỆU ĐỀ TÀI

Trong thời đại hiện nay, công nghệ thông tin đang trở thành một lực lượng mạnh mẽ, không ngừng thay đổi và cải tiến cuộc sống của chúng ta. Trong bối cảnh này, mô hình học dựa trên dự án (PBL) đã trở thành một phương pháp giảng dạy hiệu quả, đặc biệt là khi chúng ta đối mặt với thách thức giải quyết vấn đề thực tế.

Chúng em, nhóm nghiên cứu, đã chọn đề tài "Dự án viết chương trình client/server để in lịch thể kỷ" nhằm mục đích phát triển một hệ thống độc đáo, có khả năng tạo và in lịch lâu dài cho các thể kỷ khác nhau. Mục tiêu của chúng em không chỉ là tạo ra một ứng dụng chức năng, mà còn là khám phá cách tích hợp công nghệ và sáng tạo để giải quyết nhu cầu thực tế trong xã hội ngày nay.

Khả năng tiện lợi và linh hoạt của ứng dụng không chỉ giúp người dùng duy trì sự tổ chức trong cuộc sống hàng ngày mà còn mở ra những cơ hội mới trong việc tận dụng thông tin lịch sử và văn hóa. Dự án không chỉ đặt ra câu hỏi về cách chúng ta lưu giữ di sản, mà còn thách thức chúng ta suy nghĩ về cách chúng ta tương tác với thời gian hàng ngày.

Bằng cách này, chúng em hy vọng đưa ra một giải pháp tích cực cho cộng đồng, khám phá cách mà công nghệ có thể giúp cho cuộc sống hàng ngày của chúng ta trở nên dễ dàng hơn.

Và cuối cùng chúng em muốn gửi lời cảm ơn đến thầy Nguyễn Văn Nguyên đã hỗ trợ chúng em trong quá trình hoàn thành dự án. Vì còn thiếu nhiều kinh nghiệm nên sẽ không thể tránh khỏi nhiều sai sót, rất mong các thầy cô bỏ qua.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Yêu cầu đề tài

Xây dựng chương trình client-server in lịch thể kỷ sử dụng javascript. Khi người dùng ở các client nhập một năm muốn in lịch thì chương trình Server thực tính toán và trả về kết quả cho client lịch của năm đó.

1.2 Giới thiệu mô hình Client/Server

1.2.1 Cấu trúc mô hình Client/Server

Client / Server là mô hình mạng máy tính gồm có 2 thành phần chính đó là máy khách (client) và máy chủ (server). Server chính là nơi giúp lưu trữ tài nguyên cũng như cài đặt các chương trình dịch vụ theo đúng như yêu cầu của client. Ngược lại, Client bao gồm máy tính cũng như các loại thiết bị điện tử nói chung sẽ tiến hành gửi yêu cầu đến server.

Mô hình mạng Client / Server sẽ cho phép mạng tập trung các ứng dụng có cùng chức năng tại một hoặc nhiều dịch vụ file chuyên dụng. Chúng sẽ trở thành trung tâm của hệ thống. Hệ điều hành của mô hình Client/Server sẽ cho phép người dùng chia sẻ đồng thời cùng một loại tài nguyên mà không giới hạn vị trí địa lý.

Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. (Giao thức là gì thì các bạn có thể tham khảo tại đây) Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP,...

1.2.1 Máy chủ

Server được hiểu là máy chủ, thường là một hệ thống máy lớn, có bộ xử lý mạnh, có khả năng hoạt động đáng tin cậy, có khả năng lưu trữ dữ liệu lớn, nó chuyên quản lý tài nguyên (chủ yếu là cơ sở dữ liệu), cung cấp các dịch vụ mạng cho các máy khách (client) sử dụng. Bình thường nó chạy suốt thời gian thực và sẵn sàng chấp nhận các yêu cầu kết nối và các yêu cầu dịch vụ khác từ máy khách.

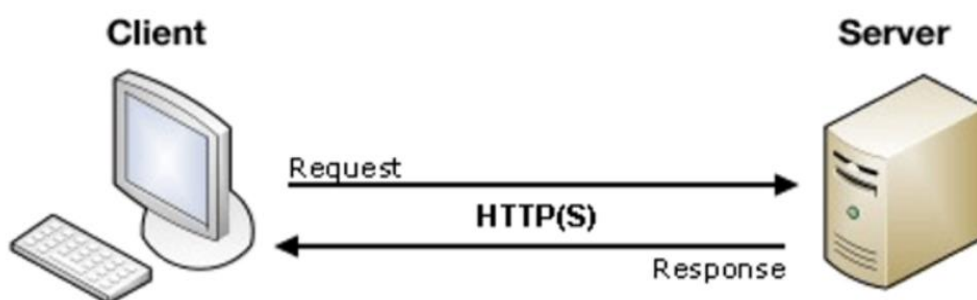
1.2.2 Máy khách

Client là máy khách, nó thường được sử dụng bởi người dùng cuối. Nó hoạt động dựa trên việc sử dụng dịch vụ mà máy server cung cấp để thực hiện các công việc mà người dùng cuối mong muốn.

1.2.3 Quy trình hoạt động

Quy trình hoạt động của mô hình này lặp lại 2 quá trình như sau:

- Client gửi yêu cầu lên server.
- Server nhận được yêu cầu thì sẽ xử lý thích hợp và trả lời lại client.



Hình 1. Quy trình hoạt động Client/Server

1.3 Giới thiệu giao thức TCP/IP

1.3.1 Giới thiệu về TCP

Giao thức TCP (Transmission Control Protocol) là giao thức hướng kết nối (connection-oriented), nó đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất theo đúng thứ tự: thiết lập kết nối, truyền dữ liệu và kết thúc kết nối.

TCP xác định cách các ứng dụng tạo kênh giao tiếp trong mạng. Ngoài ra, nó cũng quản lý cách các tin được phân thành các gói nhỏ trước khi được chuyển qua Internet và được tập hợp lại theo đúng thứ tự tại địa chỉ đến.

1.3.2 Giới thiệu về IP

IP là địa chỉ của một máy tính trên mạng, dựa vào địa chỉ IP giao thức TCP có thể truyền dữ liệu chính xác từ một máy này qua máy kia thông qua hệ thống mạng. Ở trên mạng, một máy tính sẽ có một địa chỉ IP khác nhau, từ địa chỉ IP có thể biết được máy nào trên mạng và ngược lại.

1.3.3 Giới thiệu về cổng

Port là giao thức bit 16 đứng đầu (chèn vào phần đầu header) của mỗi gói tin trong giao thức TCP, UDP hay còn gọi là cổng port, nơi quy định các tập dữ liệu riêng biệt. Port là một dạng thuật toán đã được định sẵn và mỗi máy tính cần phải có thì mới có thể nhận và gửi các gói tin đi được. Khi hoạt động Port cũng được quy đổi giống với số bit của bất kỳ một mã dữ liệu nào đó. Đơn giản thì đây giống như cánh cổng có quyền cho vào hay không với các dữ liệu muốn ra vào hệ thống máy tính.

Cần 1 địa chỉ IP, bạn xác định được một máy trong một mạng, nhưng nếu chúng chạy cùng nhiều dịch vụ khác nhau thì phải có cách để phân biệt được chúng và lúc này cần đến Port.

1.4 Giới thiệu giao thức HTTP/HTTPS

Đây là một trong những giao thức cơ bản nhất của Internet và đóng vai trò quan trọng trong việc truyền tải dữ liệu giữa máy chủ và máy khách.

HTTP hoạt động dựa trên mô hình yêu cầu/đáp ứng (request/response), trong đó:

- Yêu cầu (Request): Trình duyệt web gửi yêu cầu đến máy chủ để lấy tài nguyên cụ thể. Yêu cầu HTTP bao gồm: GET, POST, PUT, DELETE, etc.
- Đáp ứng (Response): Máy chủ phản hồi yêu cầu từ trình duyệt với một mã trạng thái và dữ liệu được yêu cầu (nếu có).

Giao thức này chủ yếu sử dụng trong việc truyền tải các trang web, hình ảnh, video và dữ liệu khác trên Internet, và cung cấp cơ chế để truyền tải dữ liệu một cách an toàn và hiệu quả.

CHƯƠNG 2. PHÂN TÍCH BÀI TOÁN

2.1 Phân tích chức năng của chương trình

Đề tài của nhóm là “Xây dựng chương trình Client/Server để in lịch thể kỷ” được chia thành 2 thành phần chính là phía Máy Chủ (Server) và phía Máy Khách (Client)

Phát triển ứng dụng theo kiến trúc Client/Server: Chương trình dựa trên mô hình kiến trúc client/server, tập trung vào việc triển khai trên nền mạng đa máy (ví dụ, mạng cục bộ...). Để đáp ứng yêu cầu này, chúng ta sẽ tuân thủ phương pháp đã được giới thiệu trong các phần trước, dựa trên các cơ sở lý thuyết đã được trình bày.

Tiếp nhận yêu cầu xem lịch của năm, tháng cụ thể và trả về thông tin lịch cần tìm: Đầu vào là tháng, năm có định dạng số, và đầu ra là thông tin lịch dưới dạng text/html trả về cho client.

2.2 Máy chủ

- **Tiếp nhận yêu cầu**

Máy chủ đón nhận các yêu cầu và lắng nghe dữ liệu để thực hiện các quy trình logic, thuật toán và truy vấn.

- **Tính toán lịch cho dữ liệu**

Máy chủ sử dụng thuật toán để tính toán lịch cho tháng, năm đã được cung cấp từ máy khách.

- **Trả kết quả về phía máy khách**

Sau khi tính toán, máy chủ gửi kết quả dưới dạng text/html (lịch cho năm đã nhập) trở lại máy khách thông qua phản hồi HTTP.

- **Bảo mật và kiểm soát lỗi**

Chương trình được thiết kế để đảm bảo tính bảo mật của dữ liệu được truyền tải giữa máy khách và máy chủ. Nếu có lỗi xảy ra trong quá trình xử lý yêu cầu hoặc tính toán lịch, chương trình phải xử lý lỗi đó một cách an toàn.

2.3 Máy khách

- **Nhập yêu cầu từ phía máy khách**

Cho phép người dùng nhập vào một năm hoặc tháng từ giao diện người dùng trên máy khách với các trường nhập liệu rõ ràng, chính xác và nhanh chóng.

- **Gửi yêu cầu đến máy chủ**

Sau khi người dùng nhập năm, máy khách gửi yêu cầu được đóng gói và gửi thông qua giao thức HTTP POST chứa thông tin cần thiết đến máy chủ.

- **Hiện thị lịch trên giao diện người dùng**

Máy khách nhận kết quả, tiến hành giải mã hoặc trích xuất dữ liệu từ phản hồi nhận được và hiển thị lịch tương ứng lên giao diện người dùng.

2.4 Thiết kế chương trình

Chương trình bao gồm 2 thành phần chính là Frontend và Backend. Quá trình triển khai dự án được chia thành hai giai đoạn chính để đảm bảo tính linh hoạt và hiệu suất của dự án:

- **Frontend:**

1. Sử dụng Reactjs để phát triển giao diện người dùng (GUI) kết hợp với các công cụ khác như Html, Css,..., giúp tạo ra các thành phần như khung nhập dữ liệu, nút bấm,...
2. Thực hiện việc gọi dữ liệu từ Backend lấy dữ liệu về lịch cần để hiển thị trên giao diện

- **Backend:**

1. Sử dụng Nodejs để phát triển chương trình máy chủ xử lý logic, thuật toán, và các sự kiện liên quan
2. Tối ưu hóa thuật toán, đảm bảo rằng quá trình xử lý thông tin diễn ra một cách hiệu quả, tăng hiệu suất

2.5 Phương pháp xử lý

2.5.1 Máy khách

Như đã trình bày khái quát quá trình hoạt động của máy khách ở trên, chương trình máy khách gồm 3 hoạt động chính

- **Nhập yêu cầu từ phía máy khách**

Chương trình sử dụng nền tảng Reactjs hiển thị giao diện cho phép người dùng tương tác nhập liệu thông qua Radio Button, Dropdown, Input Field, Button,...

Đoạn chương trình nhỏ sau đây in ra Radio Button, sử dụng Html kết với cú pháp của Reactjs.

1	<code><div></code>
2	<code> <label></code>
3	<code> <input</code>
4	<code> type="radio"</code>
5	<code> value="option1"</code>
6	<code> checked={selectedOption === 'option1'}</code>
7	<code> onChange={radioButtonChange}</code>
8	<code> /></code>
9	<code> Lịch tháng</code>

```

10     </label>
11     <br/>
12     <label>
13         <input
14             type="radio"
15             value="option2"
16             checked={selectedOption === 'option2'}
17             onChange={radioButtonChange}
18         />
19         Lịch năm
20     </label>
21     <select name="month"
22         value={month}
23         onChange={(e) =>
24             setMonth(e.target.value)}
25         className="navi-r"
26     >
27         {Array.from({ length: 12 }, (_, index) => (
28             <option key={index} value={{index + 1}.toString()}>{index + 1}
29         </option>
30         <button class="navi-r" type="button" onClick={viewMonth}>
31             Xác nhận
32         </button>
33     </div>

```

Giao diện hoàn chỉnh để nhập dữ liệu cơ bản ở máy khách sau khi bao gồm các thành phần, trường nhập liệu cần thiết trông như sau:

Hình 2. Nhập dữ liệu

Có 2 chức năng chính cho chương trình, đó là:

- Xem lịch tháng: chỉ hiển thị lịch tháng cụ thể cần xem
- Xem lịch năm: hiển thị toàn bộ 12 tháng của năm

Để biết được tính năng người dùng đang muốn sử dụng, chương trình sẽ nhận diện dựa trên tình trạng chọn nút radio

```

1  const radioButtonChange = (e) => {
2    setSelectedOption(e.target.value);
3    const option = e.target.value;
4    if(option == 'option1'){
5      setIsShowMonthCalendar(true);
6      setIsShowYearCalendar(false);
7    }else{
8      setIsShowMonthCalendar(false);
9      setIsShowYearCalendar(true);
10   }
11  };

```

Khi người dùng bấm vào một nút radio, hàm `radioButtonChange()`, sẽ được gọi kèm sự kiện, triết xuất giá trị của nút vừa được chọn và lưu trạng thái, đồng thời chỉnh chức năng hiện tại người dùng đang chọn.

Ngoài việc nhận diện bằng nút bấm, chương trình còn có thể chuyển đổi chức năng thông qua dò xét cụ thể đường dẫn mà người dùng đã sử dụng, ví dụ:

<https://lichthekey-frontend.onrender.com/?month=9&year=2013>

Ngoài đường dẫn đã cung cấp, người dùng có linh hoạt để tùy chỉnh theo nhu cầu cụ thể hoặc lựa chọn phù hợp với từng trường hợp, cụ thể:

```

1  <Routes>
2    <Route path="/">
3      <Route index element={<Calendar/>} />
4      <Route path="/xemlich" element={<Calendar/>} />
5      <Route path="*" element={<Missing/>} />
6    </Routes>
7  </Routes>

```

Có thể sử dụng chương trình theo từng “route” tức tuyến đường truy cập, ngoài đường dẫn `http://localhost:3000/`, người dùng có thêm `/xemlich` phía sau để có thể bắt đầu sử dụng

Nếu người dùng truy cập bằng bất kì đường dẫn nào khác, chương trình sẽ điều hướng tới một trang mặc định “Missing” để thông báo địa chỉ không tồn tại

```

1  const fetchCalendarText = async () => {
2    try {
3      const response = await axios.get(`/calendar?year=${year}`, {
4        responseType: 'text' });
5      setCalendarText(response?.data || '');
6    } catch (error) {
7      console.error(`Error fetching calendar text: ${error.message}`);
8    }
9  };

```

```

10
11     if(month != null){
12         try {
13             const response = await axios.get(`/calendar?month=${month}
14             &year=${year}`, { responseType: 'text' });
15             setMonthYearText(response?.data || '');
16         } catch (error) {
17             console.error(`Error fetching calendar month year: ${erro
18             r.message}`);
19         }
20     }
21 };

```

Khi người dùng nhấn nút để gửi yêu cầu đến máy chủ, chương trình trên máy khách sẽ xử lý đầu vào month và year bằng cách sử dụng hàm `URLSearchParams()`.

Nếu biến month tồn tại trong đường dẫn, chương trình sẽ hiển thị lịch của tháng đó; ngược lại, nếu không có, chương trình sẽ mặc định hiển thị lịch của năm.

Bên cạnh việc nhập tháng và năm trong các khung input, người dùng cũng có thể xác nhận để chuyển đến bước tiếp theo. Điều này giúp xác định hiển thị lịch tháng hoặc năm dựa trên thông tin người dùng cung cấp.

- **Gửi yêu cầu đến máy chủ**

Khi người dùng nhấn nút xác nhận hoặc sử dụng phím Enter, ứng dụng sẽ kích hoạt một chức năng gửi yêu cầu Get đến máy chủ, chứa thông tin về tháng và năm.

```

1  try {
2      const response = await axios.get(`/calendar?month=${month}&year=
3      ${year}`, { responseType: 'text' });
4
5      setCalendarText(response?.data || '');
6  } catch (error) {
7      console.error(`Error fetching calendar text: ${error.message}`);
8  }

```

Máy khách thực hiện truyền dữ liệu thông qua một yêu cầu GET, sử dụng thư viện axios để gửi các thông tin như tháng và năm tới đường dẫn /calendar.

Khi phía máy chủ trả về kết quả, chương trình lưu dữ liệu trả về là `response?.data` để sử dụng ở bước tiếp theo.

Nếu trường hợp máy chủ bị sự cố, khối `catch()` sẽ được gọi để tránh các lỗi hệ thống

- **Hiển thị lịch trên giao diện**

Dữ liệu nhận được từ phía Server trả về được thể hiện dưới dạng tập tin text/html và chương trình Client hiển thị ra màn hình bằng cách hiển thị trực tiếp dữ liệu html

```

1 { monthYearText ? (
2   <>
3   {isShowMonthCalendar && <span dangerouslySetInnerHTML={{
4   __html: monthYearText }} />
5   }
6   </>
7 ) : (
8   <>
9   <p>Đang lấy dữ liệu lịch tháng</p>
10  </>
11 )}

```

Đoạn mã kiểm tra các yêu cầu thỏa mãn như đã tồn tại monthYearText hay đã cho phép in lịch chưa isShowMonthCalendar. Nếu đã thỏa mãn thì sẽ xuất dữ liệu Html nhận được từ máy chủ ra trực tiếp giao diện.

Tùy vào trạng thái chức năng đang sử dụng mà chương trình in ra giao diện phù hợp

```

1 {
2   selectedOption == 'option1' &&
3   <>
4   <p style={{ marginTop: "1rem" }}>
5     <div>
6       <form name="SelectMonth" action="">
7         Tháng { ' ' }
8         <select name="month"
9           value={month}
10          onChange={(e) =>
11            setMonth(e.target.value)}
12          className="navi-r"
13          >
14           {Array.from({ length: 12 }, (_, index) => (
15             <option key={index} value={(index +
16             1).toString()}>{index + 1}</option>
17           ))}
18         </select>
19         ...
20       </form>
21     </div>
22   </p>
23   </>
24 }

```

```

25 {
26   selectedOption == 'option2' &&
27   <>
28   <p style={{ marginTop: "1rem" }}>
29     { calendarText ? (
30       <>
31         Năm { ' ' }
32         <input
33           size={15}

```



```

34         className="navi-r"
35         type="text"
36         value={inputValue}
37         onChange={(e) => setInputValue(e.target.value)}
38         onKeyDown={handleKeyPress}
39     />
40     <br/><br/>
41     {isShowYearCalendar &&<span
42 dangerouslySetInnerHTML={{ __html: calendarText }} />
43     }
44
45     </>
46   ) : (
47     <>
48       <p>Đang lấy dữ liệu lịch năm</p>
49     </>
50   )}
51 </p>
52 </>
53 }

```

Chương trình lấy dữ liệu trạng thái của chức năng người dùng đang sử dụng, rồi render ra giao diện người dùng tùy thuộc vào giá trị của biến `selectedOption`.

Nếu nó có giá trị là `'option1'`, điều này nghĩa là máy khách đang bật chức năng xem lịch tháng. Đoạn mã sẽ hiển thị một form tương ứng cho chức năng đó cho phép người dùng chọn tháng từ một danh sách dropdown.

Nếu `selectedOption` có giá trị là `'option2'`, đoạn mã sẽ hiển thị một phần giao diện khác. Nó sẽ hiển thị một ô input cho phép người dùng nhập năm và một nút bấm.

Ngoài ra, nếu đã có dữ liệu lịch cho năm đã chọn, nó sẽ hiển thị lịch tương ứng. Nếu chưa có dữ liệu, sẽ hiển thị thông báo "Đang lấy dữ liệu lịch năm".

Xem Lịch

Lịch tháng

Lịch năm

Tháng

10

Năm

2023

Xác nhận

Tháng 10/2023						
CN	T2	T3	T4	T5	T6	T7
1 17/8	2 18	3 19	4 20	5 21	6 22	7 23
8 24	9 25	10 26	11 27	12 28	13 29	14 30
15 1/9	16 2	17 3	18 4	19 5	20 6	21 7
22 8	23 9	24 10	25 11	26 12	27 13	28 14
29 15	30 16	31 17				

Hình 3. Lịch tháng

Xem Lịch

Lịch tháng

Lịch năm

Năm

2023

Năm 2023																				
tháng 1							tháng 2							tháng 3						
CN	T2	T3	T4	T5	T6	T7	CN	T2	T3	T4	T5	T6	T7	CN	T2	T3	T4	T5	T6	T7
1 10/12	2 11	3 12	4 13	5 14	6 15	7 16				1 11/1	2 12	3 13	4 14			1 10/2	2 11	3 12	4 13	
8 17	9 18	10 19	11 20	12 21	13 22	14 23	5 15	6 16	7 17	8 18	9 19	10 20	11 21	5 14	6 15	7 16	8 17	9 18	10 19	11 20
15 24	16 25	17 26	18 27	19 28	20 29	21 30	12 22	13 23	14 24	15 25	16 26	17 27	18 28	12 21	13 22	14 23	15 24	16 25	17 26	18 27
22 1/1	23 2	24 3	25 4	26 5	27 6	28 7	19 29	20 1/2	21 2	22 3	23 4	24 5	25 6	19 28	20 29	21 30	22 1/2	23 2	24 3	25 4
29 8	30 9	31 10					26 7	27 8	28 9					26 5	27 6	28 7	29 8	30 9	31 10	
tháng 4							tháng 5							tháng 6						
CN	T2	T3	T4	T5	T6	T7	CN	T2	T3	T4	T5	T6	T7	CN	T2	T3	T4	T5	T6	T7
					1 11/2				1 12/3	2 13	3 14	4 15	5 16			1 14/4	2 15	3 16		
2 12	3 13	4 14	5 15	6 16	7 17	8 18	7 18	8 19	9 20	10 21	11 22	12 23	13 24	4 17	5 18	6 19	7 20	8 21	9 22	10 23
9 19	10 20	11 21	12 22	13 23	14 24	15 25	14 25	15 26	16 27	17 28	18 29	19 1/4	20 2	11 24	12 25	13 26	14 27	15 28	16 29	17 30
16 26	17 27	18 28	19 29	20 1/3	21 2	22 3	21 3	22 4	23 5	24 6	25 7	26 8	27 9	18 1/5	19 2	20 3	21 4	22 5	23 6	24 7
23 4	24 5	25 6	26 7	27 8	28 9	29 10	28 10	29 11	30 12	31 13				25 8	26 9	27 10	28 11	29 12	30 13	
30 11																				
tháng 7							tháng 8							tháng 9						

Hình 4. Lịch năm trên

		<div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1</div> <div>11/2</div> </div> <div> <div>2 3 4 5 6 7 8</div> <div>12 13 14 15 16 17 18</div> </div> <div> <div>9 10 11 12 13 14 15</div> <div>19 20 21 22 23 24 25</div> </div> <div> <div>16 17 18 19 20 21 22</div> <div>26 27 28 29 30 31</div> </div> <div> <div>23 24 25 26 27 28 29</div> <div>3 4 5 6 7 8 9 10</div> </div> <div> <div>30</div> <div>11</div> </div> </div>	<div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2 3 4 5 6</div> <div>12/3 13 14 15 16 17</div> </div> <div> <div>7 8 9 10 11 12 13</div> <div>18 19 20 21 22 23 24</div> </div> <div> <div>14 15 16 17 18 19 20</div> <div>25 26 27 28 29 30 31</div> </div> <div> <div>21 22 23 24 25 26 27</div> <div>3 4 5 6 7 8 9</div> </div> <div> <div>28 29 30 31</div> <div>10 11 12 13</div> </div> </div>	<div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2 3</div> <div>14/4 15 16</div> </div> <div> <div>4 5 6 7 8 9 10</div> <div>17 18 19 20 21 22 23</div> </div> <div> <div>11 12 13 14 15 16 17</div> <div>24 25 26 27 28 29 30</div> </div> <div> <div>18 19 20 21 22 23 24</div> <div>1/5 2 3 4 5 6 7</div> </div> <div> <div>25 26 27 28 29 30</div> <div>8 9 10 11 12 13</div> </div> </div>	
		<div> <div>tháng 7</div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1</div> <div>14/5</div> </div> <div> <div>2 3 4 5 6 7 8</div> <div>15 16 17 18 19 20 21</div> </div> <div> <div>9 10 11 12 13 14 15</div> <div>22 23 24 25 26 27 28</div> </div> <div> <div>16 17 18 19 20 21 22</div> <div>29 30 1/6 2 3 4 5</div> </div> <div> <div>23 24 25 26 27 28 29</div> <div>6 7 8 9 10 11 12</div> </div> <div> <div>30 31</div> <div>13 14</div> </div> </div>	<div> <div>tháng 8</div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2 3 4 5</div> <div>15/6 16 17 18 19</div> </div> <div> <div>6 7 8 9 10 11 12</div> <div>20 21 22 23 24 25 26</div> </div> <div> <div>13 14 15 16 17 18 19</div> <div>27 28 29 1/7 2 3 4</div> </div> <div> <div>20 21 22 23 24 25 26</div> <div>5 6 7 8 9 10 11</div> </div> <div> <div>27 28 29 30 31</div> <div>12 13 14 15 16</div> </div> </div>	<div> <div>tháng 9</div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2</div> <div>17/7 18</div> </div> <div> <div>3 4 5 6 7 8 9</div> <div>19 20 21 22 23 24 25</div> </div> <div> <div>10 11 12 13 14 15 16</div> <div>26 27 28 29 30 1/8 2</div> </div> <div> <div>17 18 19 20 21 22 23</div> <div>3 4 5 6 7 8 9</div> </div> <div> <div>24 25 26 27 28 29 30</div> <div>10 11 12 13 14 15 16</div> </div> </div>	
		<div> <div>tháng 10</div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2 3 4 5 6 7</div> <div>17/8 18 19 20 21 22 23</div> </div> <div> <div>8 9 10 11 12 13 14</div> <div>24 25 26 27 28 29 30</div> </div> <div> <div>15 16 17 18 19 20 21</div> <div>1/9 2 3 4 5 6 7</div> </div> <div> <div>22 23 24 25 26 27 28</div> <div>8 9 10 11 12 13 14</div> </div> <div> <div>29 30 31</div> <div>15 16 17</div> </div> </div>	<div> <div>tháng 11</div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2 3 4</div> <div>18/9 19 20 21</div> </div> <div> <div>5 6 7 8 9 10 11</div> <div>22 23 24 25 26 27 28</div> </div> <div> <div>12 13 14 15 16 17 18</div> <div>29 1/10 2 3 4 5 6</div> </div> <div> <div>19 20 21 22 23 24 25</div> <div>7 8 9 10 11 12 13</div> </div> <div> <div>26 27 28 29 30</div> <div>14 15 16 17 18</div> </div> </div>	<div> <div>tháng 12</div> <div>CN T2 T3 T4 T5 T6 T7</div> <div> <div>1 2</div> <div>19/10 20</div> </div> <div> <div>3 4 5 6 7 8 9</div> <div>21 22 23 24 25 26 27</div> </div> <div> <div>10 11 12 13 14 15 16</div> <div>28 29 30 1/11 2 3 4</div> </div> <div> <div>17 18 19 20 21 22 23</div> <div>5 6 7 8 9 10 11</div> </div> <div> <div>24 25 26 27 28 29 30</div> <div>12 13 14 15 16 17 18</div> </div> <div> <div>31</div> <div>19</div> </div> </div>	

Hình 5. Lịch năm dưới

2.5.2 Máy chủ

Như đã trình bày khái quát quá trình hoạt động của máy khách ở trên, chương trình máy khách gồm 4 hoạt động chính:

- **Tiếp nhận yêu cầu**

Máy chủ có các định tuyến cụ thể dùng để điều hướng các yêu cầu tới một địa chỉ cụ thể trên máy chủ. Sử dụng Express để tạo router, chứa một endpoint ('/') và xử lý yêu cầu GET tới endpoint này.

```

1  const router = express.Router();
2  const calendarController =
3  require('../controllers/calendarController');
4
5  router.route('/calendar').get(calendarController.printSelectedYear);
6
7  module.exports = router;

```

Khi có yêu cầu GET đến địa chỉ cụ thể ('/calendar'), router này sẽ kích hoạt một hàm xử lý đã được định nghĩa, đó là hàm printSelectedYear được gọi từ module calendarController

Sau khi máy khách gửi tín hiệu, máy chủ sẽ kích hoạt hàm để thu thập dữ liệu được chuyển đến từ máy khách. Trong quá trình này, máy chủ chờ đợi tiếp nhận thông tin.

```
1  const printSelectedYear = (req, res) => {
2    setOutputSize("small");
3    res.set('Content-Type', 'text/plain');
4
5    if (req.query?.month && req.query?.year){
6      const monthYear = require("./monthYearController");
7      monthYear.printSelectedMonth(req, res);
8      return;
9    }
10   if (req.query?.year){
11     res.send(printYear(parseInt(req.query.year)));
12   }else{
13     res.send(printYear(2023));
14   }
15 }
16
```

Hàm này sẽ thực hiện việc kiểm tra xem yêu cầu có chứa thông tin về tháng hay năm không, bằng cách truy cập vào `req.query?.month` và `req.query?.year`.

Khi đã xác định được thông tin yêu cầu, chương trình sẽ quyết định gọi hàm tương ứng để thực hiện chức năng cần thiết. Nếu yêu cầu là xem lịch tháng, hàm sẽ triệu gọi `printSelectedMonth()`, còn ngược lại, hàm sẽ chạy `printYear()`.

Tùy thuộc vào dữ liệu trong tin hiệu yêu cầu, hàm này tiếp tục chuyển hướng đến mã xử lý riêng để tạo ra kết quả là lịch tháng hoặc lịch năm trong bước tiếp theo.

- **Tính toán lịch cho dữ liệu**

Trước hết, chúng ta có chức năng tính toán lịch tháng, đây là việc hiển thị một tháng cụ thể trong năm, đồng thời cũng là một phần nhỏ của chức năng lớn hơn là in toàn bộ lịch của năm.

Khi chương trình nhận được cả hai dữ liệu về tháng và năm, nó sẽ gọi một hàm để tạo ra kết quả về tháng trong năm đó. Hàm này bao gồm hai chức năng con:

1. Hàm lấy dữ liệu css
2. Hàm tạo bảng cho tháng theo html

```
1  function printMonth(mm, yy) {
2    var res = "";
3    res += printStyle();
4    res += printTable(mm, yy);
5    return res;
6  }
```

Sau đó sẽ trả về kết quả là biết res có kiểu dữ liệu text/html trực tiếp về phía máy khách để in ra màn hình, dưới đây là hình ảnh kết quả html trả về

Tháng 10/2023						
CN	T2	T3	T4	T5	T6	T7
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Hình 6. HTML trả về

Trong hàm **printStyle()**, chúng ta có chức năng tạo ra một phần CSS để cải thiện trải nghiệm người dùng, giúp giao diện trở nên thân thiện hơn và dễ sử dụng hơn.

```

1  function printStyle() {
2      var fontSize = "13pt";
3      var res = "";
4      res += '<style type="text/css">\n';
5      res += '<!--\n';
6      res += '    .tennam {text-align:center; font-size:150%; line-height
7      :120%; font-weight:bold; color:#000000; background-color: white}\n';
8      res += '    .thang {font-size: '+fontSize+'; padding:1; line-height
9      :100%; font-family:Tahoma,Verdana,Arial; table-layout:fixed}\n';
10     .....
11     res += '    .cn {text-align:left;font-size:125%;line-height
12     :100%;color:red}\n';
13     res += '-->\n';
14     res += '</style>\n';
15     return res;
16 }
```

Dữ liệu thuần CSS này sẽ là 1 phần dữ liệu được gửi theo bên trong dữ liệu đầu ra res là text/Html

Trong hàm **printTable()**, quy trình in bảng lịch tháng trong năm được thực hiện thông qua hai giai đoạn chính:

1. Tạo ra tiêu đề tháng và ngày
2. Tạo các ô chứa ngày và lịch âm

```

1  function printTable(mm, yy) {
2      var i, j, k, solar, lunar, cellClass, solarClass, lunarClass;
3      var currentMonth = getMonth(mm, yy);
4
5      if (currentMonth.length == 0) return;
6      var ld1 = currentMonth[0];
7      var emptyCells = (ld1.jd + 1) % 7;
8      var res = "";
```

```

9      res += ('<table class="tháng" border="2" cellpadding="1" cellspac
10 ing="1" width="'+PRINT_OPTS.tableWidth+'">\n');
11      res += printHead(mm, yy);
12
13      for (i = 0; i < 6; i++) {
14          res += ("<tr>\n");
15          for (j = 0; j < 7; j++) {
16              k = 7 * i + j;
17              if (k < emptyCells || k >= emptyCells + currentMonth.lengt
18 h) {
19                  res += printEmptyCell();
20              } else {
21                  solar = k - emptyCells + 1;
22                  ld1 = currentMonth[k - emptyCells];
23                  res += printCell(ld1, solar, mm, yy);
24              }
25          }
26          res += ("</tr>\n");
27      }
28      res += ('</table>\n');
29      return res;
30  }
31
32
33
34

```

Hàm `printTable()` thực hiện việc tạo ra từng bảng riêng biệt đại diện cho mỗi tháng trong một năm. Nếu chỉ in một bảng, chúng ta có thể xem lịch tháng, và khi in toàn bộ 12 bảng, chúng ta có thể xem lịch năm. Điều này có nghĩa là cả hai tính năng đều chia sẻ phần code cốt lõi.

Công đoạn đầu tiên của hàm này chuẩn bị dữ liệu để hiển thị từng ngày trong tháng và phân bố chúng theo từng ngày trong tuần.

Đồng thời, nó cũng tạo các thông tin về ngày âm tương ứng và chuẩn bị sẵn tiêu đề cho từng ngày và tháng để đặt dữ liệu ngày vào.

```

1  function getMonth(mm, yy) {
2      var ly1, ly2, tet1, jd1, jd2, mm1, yy1, result, i;
3      if (mm < 12) {
4          mm1 = mm + 1;
5          yy1 = yy;
6      } else {
7          mm1 = 1;
8          yy1 = yy + 1;
9      }
10     jd1 = jdn(1, mm, yy);
11     jd2 = jdn(1, mm1, yy1);
12     ly1 = getYearInfo(yy);
13     tet1 = ly1[0].jd;
14     result = new Array();
15     if (tet1 <= jd1) {
16         for (i = jd1; i < jd2; i++) {
17             result.push(findLunarDate(i, ly1));
18         }

```

```

19     } else if (jd1 < tet1 && jd2 < tet1) {
20         ly1 = getYearInfo(yy - 1);
21         for (i = jd1; i < jd2; i++) {
22             result.push(findLunarDate(i, ly1));
23         }
24     } else if (jd1 < tet1 && tet1 <= jd2) {
25         ly2 = getYearInfo(yy - 1);
26         for (i = jd1; i < tet1; i++) {
27             result.push(findLunarDate(i, ly2));
28         }
29         for (i = tet1; i < jd2; i++) {
30             result.push(findLunarDate(i, ly1));
31         }
32     }
33     return result;
34 }

```

Hàm `getMonth()` tạo ra thông tin về lịch âm dựa trên tháng và năm dương lịch được cung cấp, xác định các biến cần thiết như các ngày julian tương ứng với tháng và năm, thông tin về năm âm lịch.

Sau đó thực hiện việc tính toán ngày Tết âm lịch của năm được cung cấp và tạo một mảng kết quả.

Dựa vào các điều kiện, hàm sẽ tạo ra một danh sách các ngày âm lịch tương ứng với khoảng thời gian từ ngày bắt đầu đến ngày kết thúc của tháng và năm dương lịch được chỉ định.

Mỗi điều kiện trong hàm đều kiểm tra các trường hợp khác nhau của ngày Tết âm lịch và xử lý việc tạo danh sách các ngày âm lịch dựa trên các quy tắc khác nhau của lịch truyền thống.

```

1  var i, j, k, solar, lunar, cellClass, solarClass, lunarClass;
2  var currentMonth = getMonth(mm, yy);
3
4  if (currentMonth.length == 0) return;
5  var ld1 = currentMonth[0];
6  var emptyCells = (ld1.jd + 1) % 7;
7  var res = "";
8  res += ('<table class="thang" border="2" cellpadding="1" cell spacing
9  ="1" width="'+PRINT_OPTS.tableWidth+'">\n');
10 res += printHead(mm, yy);

```

Dựa trên so sánh giữa `tet1`, `jd1`, và `jd2`, mã lệnh lặp qua số ngày Julian và đẩy các ngày âm lịch vào mảng `result` rồi trả về kết quả chứa các ngày âm lịch trong khoảng thời gian tháng và năm đã chỉ định.

Biến `emptyCells` sẽ tính toán số ô trống cần được tạo ra ở phía đầu bảng để căn chỉnh lịch từ đúng ngày trong tuần với `ld1` là đại diện cho ngày âm lịch đầu tiên trong tháng hiện tại

```

1  for (i = 0; i < 6; i++) {
2      res += ("

```

Vòng lặp ngoài sẽ lặp 6 lần tương ứng 6 lần trong tuần là tối đa 1 tháng (tính cả khoảng trống của ngày tháng trước), để tạo ra từng hàng `<tr>`

Với vòng lặp trong, hàm sẽ chạy 7 lần tương ứng từng ngày trong tuần, hàm sẽ kiểm tra nếu ô đó trùng với ngày của tháng trước thì sẽ bỏ qua bằng cách in khoảng trống với `printEmptyCell()`;

Ngược lại, nếu ô đó nằm trong phạm vi tháng, hàm `printCell()` được gọi để tạo ô chứa thông tin về ngày dương lịch và ngày âm lịch tương ứng.

```

1  function printCell(lunarDate, solarDate, solarMonth, solarYear) {
2      var cellClass, solarClass, lunarClass, solarColor;
3      cellClass = "ngaythang";
4      solarClass = "t2t6";
5      lunarClass = "am";
6      solarColor = "black";
7      var dow = (lunarDate.jd + 1) % 7;
8      if (dow == 0) {
9          solarClass = "cn";
10         solarColor = "black";
11     } else if (dow == 6) {
12         solarClass = "t7";
13         solarColor = "black";
14     }
15     if (solarDate == today.getDate() && solarMonth == today.getMonth
16     () + 1 && solarYear == today.getFullYear()) {
17         cellClass = "homnay";
18     }
19     if (lunarDate.day == 1 && lunarDate.month == 1) {
20         cellClass = "tet";
21     }
22     if (lunarDate.leap == 1) {
23         lunarClass = "am2";
24     }
25     var lunar = lunarDate.day;

```



```

26     if (solarDate == 1 || lunar == 1) {
27         lunar = lunarDate.day + "/" + lunarDate.month;
28     }
29     var res = "";
30     var args = lunarDate.day + "," + lunarDate.month + "," + lunarDate.year + "," + lunarDate.leap;
31     args += ("," + lunarDate.jd + "," + solarDate + "," + solarMonth + "," + solarYear);
32     res += ('<td class="'+cellClass+'"');
33     if (lunarDate != null) res += (' title="'+getDayName(lunarDate) + '"" onClick="alertDayInfo('+args+')";');
34     res += (' <div style=color:'+solarColor+' class="'+solarClass + '""> ' + solarDate+'</div> <div class="'+lunarClass+'"">'+lunar+'</div></td>\n');
35     return res;
36 }

```

Các biến như `cellClass`, `solarClass`,... được sử dụng để quyết định lớp CSS và màu sắc của ô lịch, đại diện cho các điều kiện khác nhau mà ô lịch có thể có, chẳng hạn như ô là Chủ nhật, Thứ Bảy, hoặc ngày nhuận trong lịch âm.

Hàm sẽ thực hiện kiểm tra vị trí của ngày trong tuần để đặt lớp và màu sắc tương ứng. Nó cũng kiểm tra xem liệu ngày hiện tại có trùng với ngày được xác định không.

Nếu ngày âm là ngày đầu tiên của tháng hoặc ngày dương là ngày đầu tiên của tháng, hàm sẽ thêm lớp CSS tương ứng.

Và ngoài ra còn tạo chuỗi HTML để tạo ô lịch với tất cả các thuộc tính đã thiết lập dựa trên các điều kiện đã kiểm tra.

Còn nếu như ô đó thuộc một ngày ở tháng trước, hoặc sau, nó sẽ được bỏ trống với hàm `printEmptyCell()`

```

1  function printEmptyCell() {
2      return '<td class=ngaythang><div class=cn>&nbsp;</div> <div class=am>&nbsp;</div></td>\n';
3  }

```

Sau khi kết thúc vòng lặp xử lý các cột trong một hàng, thêm `</tr>` để kết thúc hàng hiện tại và bắt đầu hàng mới.

Trong ngữ cảnh của việc tạo lịch năm, đây như một bức tranh to lớn hơn, chứa đựng tất cả các hàm tạo lịch tháng.

```

1  function printYear(yy) {
2      var yearName = " Năm " + yy;
3      var res = "";
4      res += printStyle();
5      res += '<table class="thang" border="2" cellpadding="1" cellspacing="1" align=center>\n';

```

```

7      res += ('<tr><td colspan="3"
8      class="tennam">'+yearName+'</td></tr>\n');
9      for (var i = 1; i<= 12; i++) {
10         if (i % 3 == 1) res += '<tr>\n';
11         res += '<td>\n';
12         res += printTable(i, yy);
13         res += '</td>\n';
14         if (i % 3 == 0) res += '</tr>\n';
15     }
16     res += '<table>\n';
17     res += printFoot();
18     return res;
19 }

```

Trong thực tế, việc tạo lịch cho cả năm thực chất là quá trình tạo từng lịch tháng mười hai lần tương ứng với năm đó. Phần quan trọng của chương trình chính là việc tạo lịch cho cả năm.

- **Trả kết quả về phía máy khách**

Sau khi xử lý xong dữ liệu, chương trình sẽ trực tiếp gửi toàn bộ thông tin dưới dạng văn bản/HTML quay trở lại cho người dùng thông qua chức năng `printMonth()`.

Dữ liệu được trả về được định dạng dưới dạng text/Html.

```

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
[nodeemon] restarting due to changes...
[nodeemon] starting 'node server.js'
Server running on port 3500
GET /calendar
GET /calendar
<style type="text/css">
<!--
.tennam {text-align:center; font-size:150%; line-height:120%; font-weight:bold; color:#000000; background-color: white}
.thang {font-size: 13pt; padding:1; line-height:100%; font-family:Tahoma,Verdana,Arial; table-layout:fixed}
.tenthang {text-align:center; font-size:125%; line-height:100%; font-weight:bold; color:#330033; background-color: white;}
.navi-l {text-align:center; font-size:75%; line-height:100%; font-family:Verdana,Times New Roman,Arial; font-weight:bold; color:red; background-color: white}
.navi-r {text-align:center; font-size:75%; line-height:100%; font-family:Verdana,Arial,Times New Roman; font-weight:bold; color:#330033; background-color: white}
.ngaytuan {width:14%; text-align:center; font-size:125%; line-height:100%; color:#330033; background-color: white}
.ngaythang {background-color: rgba(255,255,255,0.1)}
.hommay {background-color:#FFF000}
.tet {background-color:#FFC99}
.am {text-align:right;font-size:75%;line-height:100%;color:blue}
.am2 {text-align:right;font-size:75%;line-height:100%;color:#004080}
.t2t6 {text-align:left;font-size:125%;color:black}
.t7 {text-align:left;font-size:125%;line-height:100%;color:green}
.cn {text-align:left;font-size:125%;line-height:100%;color:red}
-->
</style>
<table class="thang" border="2" cellpadding="1" cellspacing="1" width="420px">
<tr><td colspan="2" class="navi-l">&nbsp;&nbsp;&nbsp;</td>
<td colspan="3" class="tenthang" >Tháng 10/2023</td>
<td colspan="2" class="navi-r">&nbsp;&nbsp;&nbsp;</td></tr>
<tr onClick="alertAbout();">
<td class=ngaytuan>CN</td>
<td class=ngaytuan>T2</td>
<td class=ngaytuan>T3</td>
<td class=ngaytuan>T4</td>
<td class=ngaytuan>T5</td>
<td class=ngaytuan>T6</td>
<td class=ngaytuan>T7</td>
</tr>
<tr>
<td class="ngaythang" title="Ngày Nh\342m Th\354n, th\341ng T\342n D\u1EADu, năm Qu\375 M\343o" onClick="alertDayInfo(17,8,2023,0,2460219,1,10,2023);"><div style=
color:black class="cn">1</div> <div class="am">17/8</div></td>
<td class="ngaythang" title="Ngày Qu\375 T\u1EF5, th\341ng T\342n D\u1EADu, năm Qu\375 M\343o" onClick="alertDayInfo(18,8,2023,0,2460220,2,10,2023);"><div style=
color:black class="t2t6">2</div> <div class="am">18</div></td>
<td class="ngaythang" title="Ngày Gi\341p Ng\u1ECD, th\341ng T\342n D\u1EADu, năm Qu\375 M\343o" onClick="alertDayInfo(19,8,2023,0,2460221,3,10,2023);"><div style=
color:black class="t2t6">3</div> <div class="am">19</div></td>
<td class="ngaythang" title="Ngày \u1EA4t M\371i, th\341ng T\342n D\u1EADu, năm Qu\375 M\343o" onClick="alertDayInfo(20,8,2023,0,

```

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Các công cụ sử dụng

- ***Github***

Là hệ thống quản lý dự án và phiên bản code, dùng để quản lý và đồng bộ chương trình dễ dàng giữa các thành viên trong nhóm, giúp cùng nhau phối hợp để theo dõi tiến độ, cập nhật hiệu quả.

- ***Visual Studio Code***

Trình biên dịch, soạn code hiệu quả của Microsoft dành cho các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở.

- ***Node.js***

Là một runtime environment dùng để thực thi mã javascript, dành cho cả client và back-end. Dùng để xây dựng các ứng dụng quy mô lớn cần xử lý lượng request đồng thời lớn

- ***ThunderClient***

Công cụ cho phép chúng ta thao tác với API, mà không cần viết dòng code nào, sử dụng trong quá trình phát triển, kiểm thử. Đây là một tiện ích nhỏ gọn được tích hợp trong Visual Studio Code.

- ***Microsoft Teams, Messenger, Zalo***

Các phương tiện liên lạc, trao đổi thông tin và báo cáo tiến độ hàng tuần

3.2 Môi trường

- Hệ điều hành: Windows 10
- Javascript phiên bản ES13

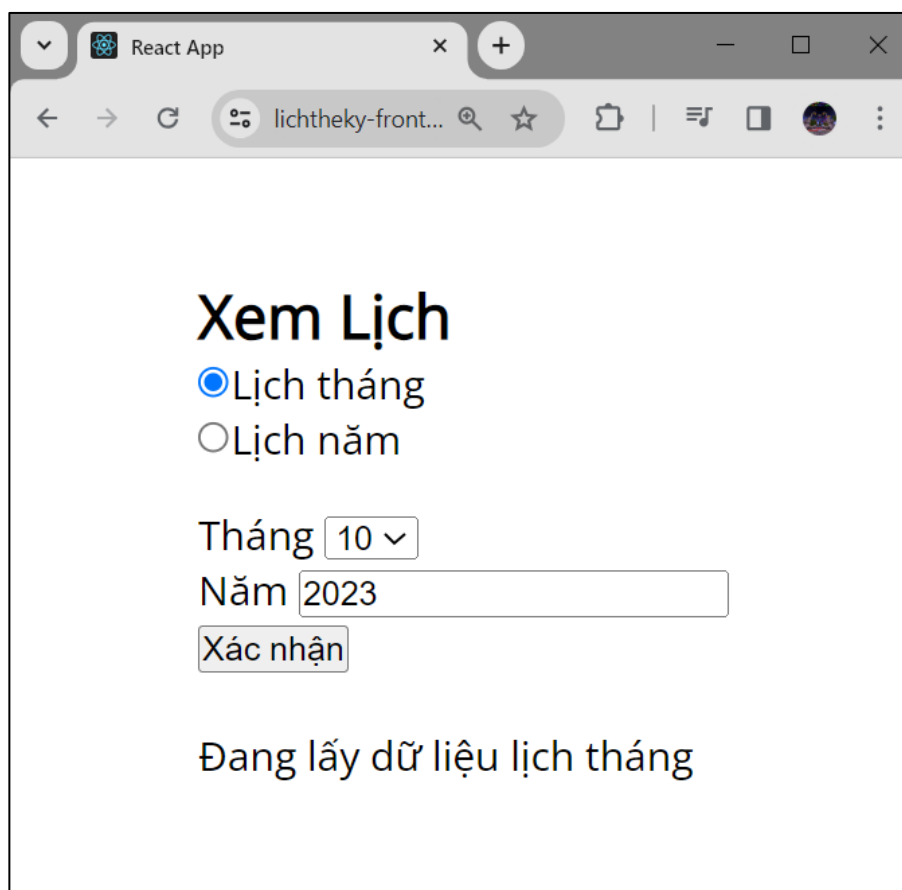
3.3 Kết quả

Chương trình đáp ứng các yêu cầu cơ bản của bài toán, cho phép in lịch thể kỷ dựa trên mô hình Client/Server. Giao diện người dùng ổn định, đáp ứng nhanh và không gặp sự cố trong quá trình sử dụng.

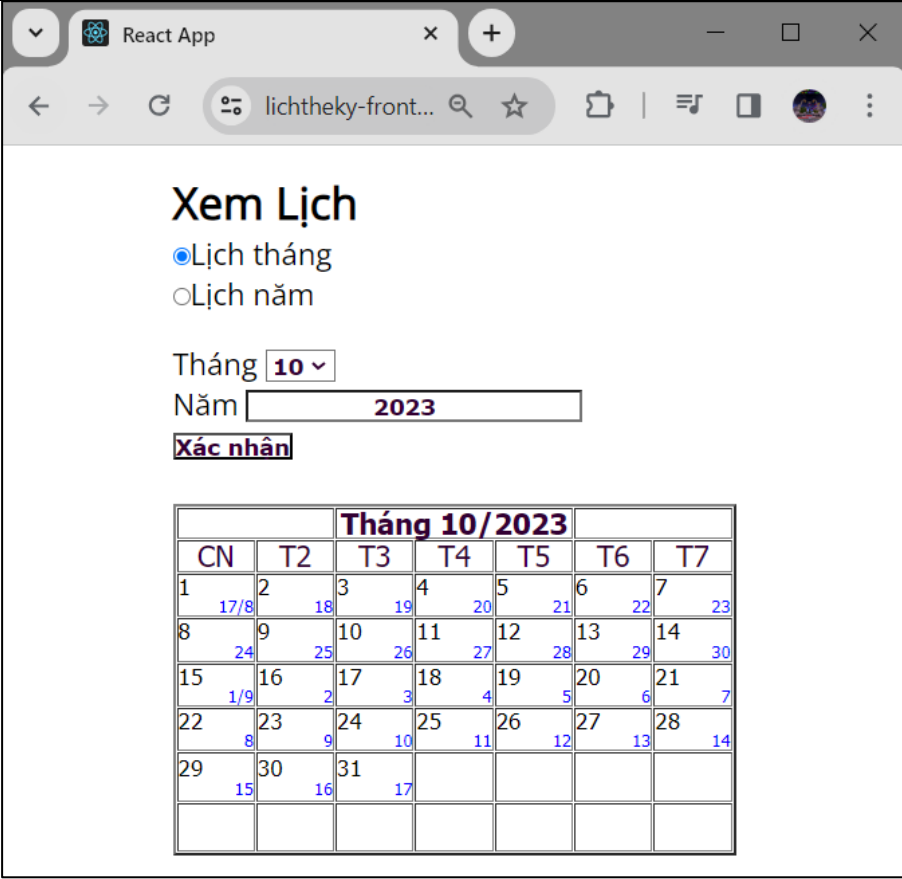
Ngoài ra, ứng dụng đã được triển khai trên Cloud và có sẵn để truy cập qua website tại địa chỉ:

<https://lichthekey-frontend.onrender.com/>

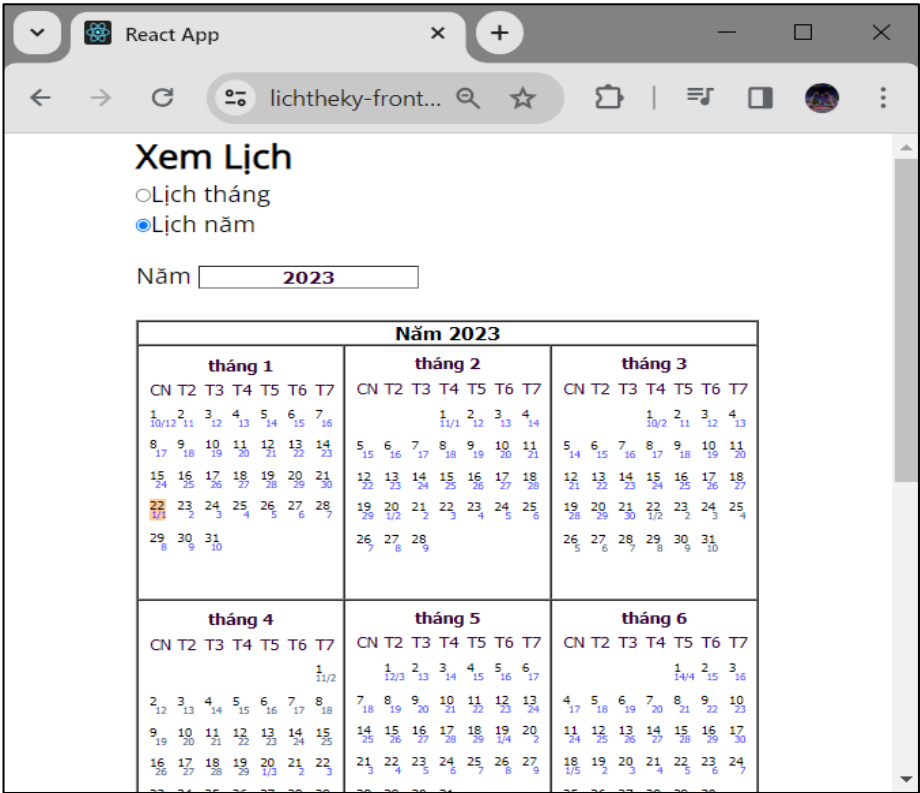
Dưới đây là toàn bộ giao diện tổng quát của cả chương trình in lịch thể kỷ với mô hình Client/Server:



Hình 7. Giao diện chính chương trình



Hình 8. Giao diện xem lịch tháng



Hình 9. Giao diện xem lịch năm

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Các kết quả thu được từ chương trình đã cơ bản hoàn thành yêu cầu cần thiết cho bài toán, cung cấp một giải pháp để xem thông tin lịch thể kỷ.

Ngoài ra, giao diện được xây dựng đã tạo ra một cái nhìn tổng quan, giúp người dùng hiểu rõ hơn về cách hoạt động, cơ chế và quá trình vận hành.

a) Đạt được

- Tốc độ nhanh, không bị xung đột khi chạy chương trình, đưa kết quả lịch với độ chính xác đến người dùng
- Thuật toán được áp dụng hợp lý, giải quyết tốt vấn đề

b) Chưa đạt được

- Khả năng đáp ứng nhu cầu của người dùng còn hạn chế, chưa đa dạng và linh hoạt trong việc cung cấp các tính năng.
- Giao diện vẫn giữ độ đơn giản, chưa thể tối ưu hóa trải nghiệm người dùng với một hình ảnh đẹp mắt và thu hút hơn.
- Tính năng hiện tại vẫn còn hạn chế, khá cứng nhắc và thỉnh thoảng gặp phải các lỗi trong quá trình sử dụng

c) Hướng phát triển

Đề tài này đã giúp chúng em có cái nhìn tổng quan hơn về hoạt động của mô hình Client/Server và cách áp dụng nó vào giải quyết một bài toán cụ thể, như việc xây dựng chương trình in lịch thể kỷ.

Tuy đã hoàn thành dự án, nhưng trong phạm vi của đề tài, sản phẩm vẫn còn khá đơn giản về giao diện và chưa khai thác đầy đủ các tính năng mở rộng và phức tạp. Một số hướng phát triển tiềm năng trong tương lai có thể bao gồm:

- Phát triển giao diện ứng dụng với thiết kế thẩm mỹ, tương tác người dùng tốt hơn và thân thiện hơn.
- Cải thiện thuật toán để tối ưu hóa tốc độ và hiệu suất, từ đó dễ dàng mở rộng và áp dụng cho các bài toán phức tạp hơn.
- Tăng cường bảo mật để đối phó với các cuộc tấn công và rủi ro từ bên ngoài.

TÀI LIỆU THAM KHẢO

- [1] Trần Hồ Thủy Tiên, *Bài giảng Nguyên lý hệ điều hành*, Khoa CNTT Trường Đại học Bách khoa
- [2] Nguyễn Thế Xuân Ly, *Giáo trình Mạng máy tính*, Khoa CNTT Trường Đại học Bách khoa
- [3] Nguyễn Gia Định, *Nguyên lý hệ điều hành*, NXB Khoa học và kỹ thuật, 2004
- [4] Dave Gray, *Node.js Full Course for Beginners / Complete All-in-One ...*, 2021
- [5] Mario Casciaro, *Node.js Design Patterns*, Packt Publishing Ltd, 2016

PHỤ LỤC

[1] Mã nguồn chương trình Client, <https://github.com/pbl4lichthekey/pbl4-lichthekey-frontend>

[2] Mã nguồn chương trình Server, <https://github.com/pbl4lichthekey/pbl4-lichthekey-backend>

[3] Chương trình toàn diện online, <https://lichthekey-frontend.onrender.com>