



2023-2024

Web Programming  
Applications

# TIMETABLE GENERATOR

Documentation





Presented For:

**Dr. Mona Nashaat**

Presented By:

**EI-NHRR Group**

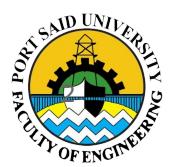
Nour Abuelenin  
Hla Dawoud  
Rawan Zangir  
Rovan Wael





## Table of Content

<b>Introduction</b>	<b>2</b>
Technologies Used	2
Key Features & Functionalities	2
<b>Project Overview</b>	<b>3</b>
Overall Structure	3
Frontend - User Interface	5
Backend - Project's Backbone	7
<b>Project Functionality</b>	<b>8</b>
<b>Implementation</b>	<b>9</b>
HTML	9
Login Page	9
Main Page	11
JavaScript	20
LoginPage	20
Main Page	22
Server	29
SQLite3	34
<b>Challenges</b>	<b>35</b>
<b>Conclusion</b>	<b>35</b>
Major Functional Improvements for the Timetable Generator	35



## Introduction

Faculty members at Port Said University often struggle to create conflict-free timetables due to the multitude of courses offered and complex schedules. This can lead to stress, frustration, and even course overload.

A web-based timetable generator specifically designed for the Faculty of Engineering, it is a friendly tool that will simplify course scheduling and ensure a conflict-free semester.

## Technologies Used

- |               |             |
|---------------|-------------|
| 1. Javascript | 4. SQLite 3 |
| 2. HTML       | 5. Node.js  |
| 3. CSS        | 6. Express  |

## Key Features & Functionalities

1. **Intuitive Interface:** The platform will feature a user-friendly interface where faculty members can easily select the major, year, and semester of the timetable.
2. **Course Database Integration:** The timetable generator will be integrated with the university's course database, automatically reserving the lecture hall at the selected time slot to prevent double booking.
3. **Easy Scheduling:** Courses can be placed in their preferred time slots, visually building the most suitable timetable.
4. **Conflict Detection and Resolution:** The generator will identify and highlight any potential time conflicts, allowing students to adjust their selections for a conflict-free schedule.



## Project Overview

### Overall Structure

The diagram below shows the project's files and folders organized in a hierarchy. Here's a breakdown of the main components:

- **Timetable Generator:** This folder likely contains the main HTML files for the application.
  - **Main:** This file contains the main HTML code.
  - **Login:** This file contains the code related to user login functionality.
- **Assets:** This folder holds static assets used by the application, such as:
  - **CSS:** Files containing styles for the user interface (UI).
  - **JS:** Files containing JavaScript code for the application's logic and interactivity.
  - **Images:** Images used in the UI.
- **SQL:** This folder contains the project's database files, used to store and manage timetable data.
- **Node-express:** This suggests the project might use Node.js and Express framework for the backend server functionality.

Overall, the structure suggests a typical web application layout with separate folders for the frontend (HTML, CSS, JS) and backend (Node.js, Express, Database).

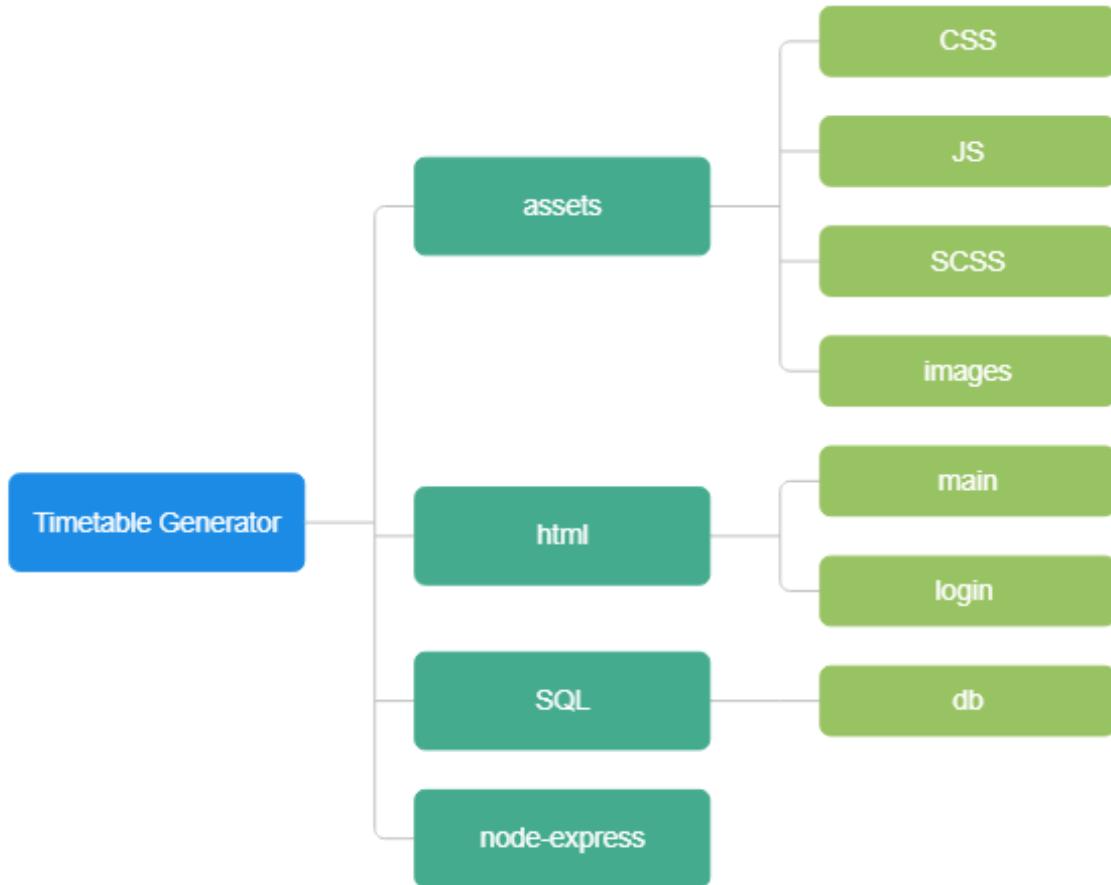


Figure (1) - Project's Directory



## Frontend - User Interface

Port Said University  
Faculty of Engineering

# Timetable Generator

Hii I'm here to help you Generate the year schedule without conflict.

Generate

Home   About   Course Specs.

...

Login

Username: Enter username

Password: Enter password

Login

Add New Course

Day: Saturday

Time Interval: 8:30 AM - 10:00 AM

Course Name:

Instructor Name:

Course Type: Lecture

Room Number:

Add Course

Please select options from the form.

Major: Select Major

Year: Select Year

Semester: Select Semester

Submit

Clear Selection

Edit/Delete Course

Course Name:

Instructor Name:

Course Type: Lecture

Room Number:

Edit Course

Delete Course

**Add New Course**

Day:

Time Interval:

Course Name:

Instructor Name:

Course Type:

Room Number:

**Add Course**

**Major - Electrical Power Engineering, Year - Fourth, Semester - First Semester**

Time/Day	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday
8:30 AM - 10:00 AM						
10:15 AM - 11:45 AM			control-2 Instructor: Kamel Lecture Room: 515			
12:15 PM - 1:45 PM			lab-control Instructor: omar Practical Lab Room: 12			
02:00 PM - 3:30 PM						
03:30 PM - 5:00 PM						

Print Timetable  Export Timetable

POWERED BY EL NHRR GROUP

Figure (2) - User Interface



## Backend – Project's Backbone

### 1. Securing User Access: Our API-Driven Authentication

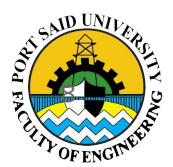
To ensure data privacy and protect user schedules, we've implemented a robust authentication system that leverages a dedicated API endpoint.

1. **User initiates login:** When a user enters their credentials on the login page, a POST request is sent to the /api/login/ endpoint.
2. **API validates data:** The API meticulously validates the incoming username and password, ensuring all fields are present and secure.
3. **Database query:** If validation passes, the API queries the database to verify the credentials against stored user information.
4. **Authentication response:** Upon successful authentication, the API sends a JSON response with a success message, granting access to the timetable generator. If authentication fails, appropriate error messages guide the user.
5. **Cross-Origin Resource Sharing (CORS):** To facilitate seamless communication between different web origins, the API includes CORS headers, expanding its integration potential.

This API-based authentication system establishes a secure foundation, ensuring that only authorized users can access and manage their personal timetables.

### 2. Timetable Data at Your Fingertips: The /api/timetable/ Endpoint

1. **Course Creation:** This API endpoint empowers users to seamlessly add courses to their timetables, ensuring a personalized schedule.
2. **Comprehensive Data:** It accepts a wide array of course details, including major, year, semester, day, time, course name, room number, course type, and instructor name, for a well-structured timetable.
3. **Database Integration:** It effectively stores course information within the database for persistent access and management.
4. **Clear Responses:** It provides informative success or error messages to guide the user experience effectively.
5. **CORS Compatibility:** It supports Cross-Origin Resource Sharing, enabling broader integration possibilities.



This API serves as the backbone for building and managing personalized timetables, ensuring a user-friendly and efficient experience.

### 3. Unveiling the Schedule: The /api/showTimetable/ Endpoint

1. **Timetable Access:** This API endpoint empowers users to effortlessly retrieve their personalized timetables, ensuring effortless schedule visibility.
2. **User-Driven Retrieval:** It requires users to provide their major, year, and semester for accurate timetable fetching, upholding data privacy and specificity.
3. **Database Interaction:** It efficiently queries the database to extract the relevant timetable information based on the provided criteria.
4. **JSON Delivery:** It delivers the retrieved timetable data as a well-structured JSON response, enabling seamless integration within the frontend.
5. **Clear Error Handling:** It provides informative error messages in case of retrieval issues, ensuring a smooth user experience.

This API serves as a gateway to accessing and displaying user-specific timetables, fostering efficient schedule management and clarity.

## Project Functionality

This code primarily constructs a dynamic webpage that enables users to generate and export their personalized academic timetables. It achieves this through a user-friendly interface where only authorized users can select the major, year of study, and semester.

Upon submitting their choices, the code fetches relevant course data and constructs a visually appealing timetable within a table format.

To enhance user experience, it offers the valuable option to export this timetable directly into a Microsoft Excel file for offline access and management.

Additionally, the webpage incorporates navigation elements for easy exploration of faculty resources and course specifications, as well as smooth scrolling effects for a seamless visual experience.

# Implementation

## HTML

### Login Page

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>
      PSU - Faculty of Engineering Timetable Generator Authentication Page
    </title>
    <link rel="icon" href="assets\img\logo2.png" />

    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css"
      integrity="sha512-78i+FfQ3Bs8T+yXT3YP8qOcWcuZdM5z85cGbFY8p9u+sBOQjQv5vExyONQAqG46tUX2sJaCMG5q/LtPaRtP65w=="
      crossorigin="anonymous"
      referrerPolicy="no-referrer-policy"
    />
    <link rel="stylesheet" href="loginstyle.css" />
  </head>
  <body>
    <header>
      
    </header>
    <main>
      <form action="/api/login/" method="POST" id="loginForm">
        <h2>Login</h2>
        <label for="username">Username:</label>

```

```
<input  
    type="text"  
    id="username"  
    name="username"  
    placeholder="Enter username"  
/>  
<label for="password">Password:</label>  
<input  
    type="password"  
    id="password"  
    name="password"  
    placeholder="Enter password"  
/>  
<i id="eye-icon" class="fas fa-eye"></i>  
<button type="submit">Login</button>  
</form>  
</main>  
<script> ... </script>  
</body>  
</html>
```



## Main Page

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <!--===== BOXICONS =====-->
    <link
      href="https://unpkg.com/boxicons@2.1.2/css/boxicons.min.css"
      rel="stylesheet"
    />

    <!--===== CSS =====-->
    <link rel="stylesheet" href="assets/css/styles.css" />

    <title>PSU - Faculty of Engineering Timetable Generator</title>
    <link rel="icon" href="assets\img\logo2.png" />
    <title>Export Table to Excel</title>
    <script
      src="https://cdn.jsdelivr.net/npm/table2excel@1.1.2/dist/table2excel.min.js"><
    /script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/exceljs/4.4.0/exceljs.js"
      integrity="sha512-KnRSGPI3rrfonYItBkenM6vyGmetr9uQViDSOb39QLvXt7EoqTn/g+pubMb7ZW9cNMNeXTIMr3utPLEf28JqiQ=="
      crossorigin="anonymous"
      referrerPolicy="no-referrer"
    ></script>
  </head>
  <body>
    <!--===== HEADER =====-->
    <header class="header">
      <nav class="nav container">
        <a href="#" class="nav__logo">
          
  Port Said University <br />
  Faculty of Engineering
</a>

<div class="nav__menu" id="nav-menu">
  <ul class="nav__list">
    <li class="nav__item">
      <a
        href="http://eng.psu.edu.eg/language/en/?lang=en"
        target="_blank"
        class="nav__link"
      >Home</a>
    </li>
    <li class="nav__item">
      <a
        href="http://eng.psu.edu.eg/language/en/about/"
        target="_blank"
        class="nav__link"
      >About</a>
    </li>
    <li class="nav__item">
      <a
        href="http://eng.psu.edu.eg/language/en/undergraduates/courses-specifications-undergraduate/?lang=en"
        target="_blank"
        class="nav__link"
      >Course Specs.</a>
    </li>
  </ul>

```

```

<div class="nav__close" id="nav-close">
  <i class="bx bx-x"></i>
</div>
</div>

<!-- Toggle button -->
<div class="nav__toggle" id="nav-toggle">
  <i class="bx bx-grid-alt"></i>
</div>
</nav>
</header>

<!--===== MAIN =====-->
<main class="main">
  <!--===== HOME =====-->
  <section class="home">
    <div class="home__container container">
      <div class="home__data">
        <span class="home__subtitle"></span>
        <h1 class="home__title">Timetable Generator</h1>
        <p class="home__description">
          Hi! I'm here to help you Generate <br />
          the year schedule without conflict.
        </p>
        <a class="home__button" onclick="openPopup()"> Generate </a>
      </div>
    </div>

    <div class="home__img">
      
      <div class="home__shadow"></div>
    </div>
  </div>

  <footer class="home__footer">
    <div class="main__action">
      <a class="main__scroll" href="#section1">
        <div class="main__scroll-box">
          <svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">

```

```

<path d="M0 0h24v24H0z" fill="none"></path>
<path
      d="M11.9997 13.1716L7.04996      8.22186L5.63574
9.63607L11.9997 16L18.3637 9.63607L16.9495 8.22186L11.9997 13.1716Z"
      fill="rgba(28,28,30,1)"
    ></path>
  </svg>
</div>

      <span class="main__scroll-text">Scroll</span>
    </a>
  </div>
</footer>
</section>
</main>

<===== SCROLLREVEAL =====>
<script src="assets/js/scrollreveal.min.js"></script>

<===== MAIN JS =====>
<script src="assets/js/main.js"></script>
<nav></nav>
<div id="popup" style="display: none">
  <a
    href="#"
    onclick="closePopup()"
    style="position: absolute; right: 9px; top: 9px"
    >X</a>
  >
  <h2>Please select options from the form.</h2>
  <form id="selectionForm">
    <label for="major">Major:</label>
    <select id="major">
      <option value="">Select Major</option>
      <option value="Electrical Power Engineering">
        Electrical Power Engineering
      </option>
      <option value="Computer and Control Engineering">
        Computer and Control Engineering
      </option>
    </select>
  </form>
</div>

```

```

        Computer and Control Engineering
    </option>
    <option value="Electronics and Communication Engineering">
        Electronics and Communication Engineering
    </option>
</select>
<label for="year">Year:</label>
<select id="year">
    <option value="">Select Year</option>
    <option value="First">First</option>
    <option value="Second">Second</option>
    <option value="Third">Third</option>
    <option value="Fourth">Fourth</option>
</select>
<label for="semester">Semester:</label>
<select id="semester">
    <option value="">Select Semester</option>
    <option value="First Semester">First</option>
    <option value="Second Semester">Second</option>
</select>
<br />
<button href="#section1" type="submit">Submit</button>
</form>
<button id="clearButton">Clear Selection</button>
</div>

<!--===== TABLE =====-->
<div
    style="
        display: flex;
        align-items: baseline;
        justify-content: space-evenly;
        margin-top: 20px;
        flex-direction: row-reverse;
    "
>
    <div style="display: flex; flex-direction: column; align-items: flex-end">

```

```

<div id="timetable" class="timetable">
    <h2 id="section1">
        
        <br />
        <p id="userSelectionHeader">Please select options from the
form.</p>
    </h2>
    <table class="containert" id="containert">
        <thead>
            <tr>
                <th>Time/Day</th>
                <th>Saturday</th>
                <th>Sunday</th>
                <th>Monday</th>
                <th>Tuesday</th>
                <th>Wednesday</th>
                <th>Thursday</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th id="time">8:30 AM - 10:00 AM</th>
                <td class="time-slot" id="saturday-830-10"></td>
                <td class="time-slot" id="sunday-830-10"></td>
                <td class="time-slot" id="monday-830-10"></td>
                <td class="time-slot" id="tuesday-830-10"></td>
                <td class="time-slot" id="wednesday-830-10"></td>
                <td class="time-slot" id="thursday-830-10"></td>
            </tr>
            <tr>
                <th id="time">10:15 AM - 11:45 AM</th>
                <td class="time-slot" id="saturday-1015-1145"></td>
                <td class="time-slot" id="sunday-1015-1145"></td>
                <td class="time-slot" id="monday-1015-1145"></td>
            </tr>
        </tbody>
    </table>

```

```

<td class="time-slot" id="tuesday-1015-1145"></td>
<td class="time-slot" id="wednesday-1015-1145"></td>
<td class="time-slot" id="thursday-1015-1145"></td>
</tr>
<tr>
  <th id="time">12:15 PM - 1:45 PM</th>
  <td class="time-slot" id="saturday-1215-145"></td>
  <td class="time-slot" id="sunday-1215-145"></td>
  <td class="time-slot" id="monday-1215-145"></td>
  <td class="slot" id="tuesday-1215-145">
    
  </td>
  <td class="time-slot" id="wednesday-1215-145"></td>
  <td class="time-slot" id="thursday-1215-145"></td>
</tr>
<tr>
  <th id="time">02:00 PM - 3:30 PM</th>
  <td class="time-slot" id="saturday-2-330"></td>
  <td class="time-slot" id="sunday-2-330"></td>
  <td class="time-slot" id="monday-2-330"></td>
  <td class="time-slot" id="tuesday-2-330"></td>
  <td class="time-slot" id="wednesday-2-330"></td>
  <td class="time-slot" id="thursday-2-330"></td>
</tr>
<tr>
  <th id="time">03:30 PM - 5:00 PM</th>
  <td class="time-slot" id="saturday-330-5"></td>
  <td class="time-slot" id="sunday-330-5"></td>
  <td class="time-slot" id="monday-330-5"></td>
  <td class="time-slot" id="tuesday-330-5"></td>
  <td class="time-slot" id="wednesday-330-5"></td>
  <td class="time-slot" id="thursday-330-5"></td>
</tr>
</tbody>
</table>
</div>
<div style="display: flex; justify-content: flex-end">

```

```

<button
    style="width: 250px; margin-inline: 5px"
    type="button"
    id="printBtn"
>
    Print Timetable
</button>
<button
    style="width: 250px; margin-inline: 5px"
    type="button"
    id="exportBtn"
>
    Export Timetable
</button>
</div>
</div>
<div>
    <form action="/api/timetable/" id="addClass" method="POST">
        <h3>Add New Course</h3>
        <label for="day">Day:</label>
        <select id="addClassDay">
            <option value="Saturday">Saturday</option>
            <option value="Sunday">Sunday</option>
            <option value="Monday">Monday</option>
            <option value="Tuesday">Tuesday</option>
            <option value="Wednesday">Wednesday</option>
            <option value="Thursday">Thursday</option>
        </select>
        <label for="time">Time Interval:</label>
        <select id="addClassTime">
            <option value="830-10">8:30 AM - 10:00 AM</option>
            <option value="1015-1145">10:15 AM - 11:45 AM</option>
            <option value="1215-145">12:15 PM - 1:45 PM</option>
            <option value="2-330">02:00 PM - 3:30 PM</option>
            <option value="330-5">03:30 PM - 5:00 PM</option>
        </select>
        <label for="courseName">Course Name:</label>
        <input type="text" id="courseName" required />
    </form>
</div>

```

```

<label for="instructorName">Instructor Name:</label>
<input type="text" id="instructorName" />
<label for="courseType">Course Type:</label>
<select id="addCourseType">
    <option value="Lecture">Lecture</option>
    <option value="Practical Lab">Practical Lab</option>
</select>
<label for="roomNumber">Room Number:</label>
<input type="text" id="roomNumber" />
<button type="submit">Add Course</button>
</form>
</div>
<div id="pop-up" class="pop-up">
<form>
    <h3>Edit/Delete Course</h3>
    <label for="courseName">Course Name:</label>
    <input type="text" id="courseName" required />
    <label for="instructorName">Instructor Name:</label>
    <input type="text" id="instructorName" />
    <label for="courseType">Course Type:</label>
    <select id="addCourseType">
        <option value="Lecture">Lecture</option>
        <option value="Practical Lab">Practical Lab</option>
    </select>
    <label for="roomNumber">Room Number:</label>
    <input type="text" id="roomNumber" />
    <button type="submit" id="editButton">Edit Course</button>
    <button id="deleteButton">Delete Course</button>
</form>
</div>
</div>
<script> ... </script>
<footer class="main__footer">
    <span>POWERED BY EL NHRR GROUP</span>
</footer>
</body>
</html>

```



## JavaScript

### LoginPage

```
<script
src="https://unpkg.com/whatwg-fetch@3.6.2/dist/fetch.umd.js"></script>

<script>
  const form = document.getElementById("loginForm");
  const passwordInput = document.getElementById("password");
  const eyeIcon = document.getElementById("eye-icon");

  eyeIcon.addEventListener("click", function () {
    if (passwordInput.type === "password") {
      passwordInput.type = "text";
      eyeIcon.classList.remove("fas fa-eye");
      eyeIcon.classList.add("fas fa-eye-slash");
    } else {
      passwordInput.type = "password";
      eyeIcon.classList.remove("fas fa-eye-slash");
      eyeIcon.classList.add("fas fa-eye");
    }
  });

  form.addEventListener("submit", (event) => {
    event.preventDefault(); // prevents default form submission

    const username = document.getElementById("username").value;
    const password = document.getElementById("password").value;

    // Send POST request with username and password data
    fetch("http://localhost:3000/api/login/", {
      method: "POST",
      body: JSON.stringify({ username, password }),
      headers: {
        "Content-Type": "application/json",
      },
    })
    .then((response) => response.json())
  });
}
```

```
.then((data) => {
  if (data.message === " Authorized success") {
    // User is authorized, redirect to desired page
    window.location.replace("main.html");
  } else {
    // User is not authorized, show error message
    alert("Sorry, you are not authorized to edit the Timetable.");
  }
})
.catch((error) => console.error(error));
});
```

</script>



## Main Page

```
<script>

    function openPopup() {
        document.getElementById("popup").style.display = "block";
    }

    function closePopup() {
        event.preventDefault();
        // Hide the form using its ID
        document.getElementById("popup").style.display = "none";
    }

    const selectionForm = document.getElementById("selectionForm");
    const userSelectionHeader = document.getElementById(
        "userSelectionHeader"
    );

    selectionForm.addEventListener("submit", (event) => {
        event.preventDefault();

        const major = document.getElementById("major").value;
        const year = document.getElementById("year").value;
        const semester = document.getElementById("semester").value;

        if (!major || !year || !semester) {
            userSelectionHeader.textContent =
                "Please select options from all fields.";
        } else {
            userSelectionHeader.textContent = `Major - ${major}, Year - ${year},
Semester - ${semester}`;
            document.getElementById("popup").style.display = "none";
        }
    });

    const timeSlots = document.querySelectorAll(".time-slot");
    const form = document.getElementById("addClass");

    form.addEventListener("submit", (event) => {
```

```

event.preventDefault();

const major = document.getElementById("major").value;
const year = document.getElementById("year").value;
const semester = document.getElementById("semester").value;
const day = document.getElementById("addClassDay").value;
const time = document.getElementById("addClassTime").value;
const courseName = document.getElementById("courseName").value;
const instructorName =
document.getElementById("instructorName").value;
const courseType = document.getElementById("addCourseType").value;
const roomNumber = document.getElementById("roomNumber").value;
if (
    !day ||
    !time ||
    !courseName ||
    !instructorName ||
    !courseType ||
    !roomNumber
) {
    alert("Please fill out all fields");
    return;
}

const timeSlotElement = document.getElementById(
    day.toLowerCase() + "-" + time
);

if (day.toLowerCase() === "tuesday" && time === "1215-145") {
    alert("Student Activity Time!");
    return; // Prevent further execution
}

if (timeSlotElement && timeSlotElement.textContent.trim()) {
    alert(
        "Selected time slot is already occupied. Please choose another day
or time interval."
    );
}

```

```

    return; // Prevent further execution
}

if (timeSlotElement && !timeSlotElement.classList.contains("slot")) {
  const timeSlotElementContent = `
    <p>${courseName}</p>
    <p>${instructorName} ? `Instructor: ${instructorName}` :
    ""}</p>
    <p>${courseType}</p>
    <p>${roomNumber} ? `Room: ${roomNumber}` : ""}</p>
  `;
  timeSlotElement.innerHTML = timeSlotElementContent;
} else {
  console.warn(
    "Selected time slot is already occupied. Please choose another."
  );
}
fetch("http://localhost:3000/api/timetable/", {
  method: "POST",
  body: JSON.stringify({
    major,
    year,
    semester,
    day,
    time,
    courseName,
    roomNumber,
    courseType,
    instructorName,
  }),
  headers: {
    "Content-Type": "application/json",
  },
})
.then((response) => response.json())
.then((data) => {
  if (data.message === "Course Added Successfully!") {
    alert("Course Added Successfully!");
  }
})

```



```
        } else {
            alert(
                "Selected time slot is already occupied. Please choose
another!"
            );
        }
    })
    .catch((error) => console.error(error));
};

const popUpElement = document.getElementById("pop-up");

timeSlots.forEach((slot) => {
    slot.addEventListener("click", (event) => {
        const courseName = slot.querySelector("p:first-child").textContent;
        console.log("Course Name:", courseName);

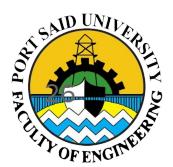
        const instructorName = slot
            .querySelector("p:nth-child(2)")
            ?.textContent?.split(" ")[1];
        console.log("Instructor Name:", instructorName);

        const courseType = slot.querySelector("p:nth-child(3)").textContent;
        console.log("Course Type:", courseType);

        const roomNumber = slot
            .querySelector("p:nth-child(4)")
            ?.textContent?.split(":")[1];
        console.log("Room Number:", roomNumber);

        const courseNameInput = document.getElementById("courseName");
        const instructorNameInput =
document.getElementById("instructorName");
        const addCourseTypeInput = document.getElementById("addCourseType");
        const roomNumberInput = document.getElementById("roomNumber");

        courseNameInput.value = courseName;
        instructorNameInput.value = instructorName;
    });
});
```



```
addCourseTypeInput.value = courseType;
roomNumberInput.value = roomNumber;

const originalColor = slot.style.backgroundColor;
popUpElement.style.display = "block";

const editButton = document.getElementById("editButton");

editButton.addEventListener("click", (event) => {
    event.preventDefault();
    // Fill input fields with new data
    const courseName = document.getElementById("courseName").value;
    const instructorName =
        document.getElementById("instructorName").value;
    const courseType = document.getElementById("addCourseType").value;
    const roomNumber = document.getElementById("roomNumber").value;

    if (!courseName || !instructorName || !courseType || !roomNumber)
    {
        alert("Please fill out all fields");
        return;
    }
    console.log("courseName", courseName);
    console.log("instructorName", instructorName);
    console.log("courseType", courseType);
    console.log("roomNumber", roomNumber);

    popUpElement.style.display = "none";
}) ;

const deleteButton = document.getElementById("deleteButton");
deleteButton.addEventListener("click", (event) => {
    // Remove data from slot
    slot.innerHTML = "";
    slot.removeAttribute("title");
    popUpElement.style.display = "none";
});
const cellContent = `
```

```

<p>${courseName}</p>
<p>${instructorName} ? `Instructor: ${courseType}` : ""</p>
<p>${addCourseType}</p>
<p>${roomNumber} ? `Room: ${roomNumber}` : ""</p>;
// Update slot content with edited information
slot.innerHTML = cellContent;
});
});

const exportBtn = document.getElementById("exportBtn");
const timetableEl = document.querySelector(".containert"); // Change the
selector to target the table directly

exportBtn.addEventListener("click", () => {
  const workbook = new ExcelJS.Workbook();
  const worksheet = workbook.addWorksheet("New Sheet");

  // Get table data
  const rows = timetableEl.querySelectorAll("tr"); // Select all rows
directly
  const headers = Array.from(
    timetableEl.querySelectorAll("thead th")
  ).map((cell) => cell.textContent); // Select headers from the first
row of the table head

  // Add headers to worksheet
  worksheet.getRow(1).values = headers;
  // Populate worksheet with data
  for (let i = 2; i <= rows.length + 1; i++) {
    const cells = rows[i - 2].querySelectorAll("td");
    worksheet.getRow(i).values = Array.from(cells).map(
      (cell) => cell.textContent
    );
  }
  // Save workbook as Excel file
  const fileName = "timetable.xlsx";
  // Use a Blob and a URL to download the file in a browser
  workbook.xlsx.writeBuffer().then((buffer) => {

```



```
const blob = new Blob([buffer], {  
    type:  
        "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",  
});  
  
const url = URL.createObjectURL(blob);  
const downloadLink = document.createElement("a");  
downloadLink.href = url;  
downloadLink.download = "timetable.xlsx";  
document.body.appendChild(downloadLink); // Append the link to the  
body  
  
downloadLink.click();  
document.body.removeChild(downloadLink); // Remove the link after  
clicking  
URL.revokeObjectURL(url); // Clean up the URL object  
alert("Timetable exported successfully!");  
});  
});  
  
const printBtn = document.getElementById("printBtn");  
printBtn.addEventListener("click", printTable);  
  
function printTable() {  
    const table = document.getElementById("containert");  
    const newWindow = window.open();  
    newWindow.document.write(table.outerHTML);  
    newWindow.document.close();  
    newWindow.print();  
    newWindow.close();  
}  
</script>
```



## Server

```
const express = require("express");
const sqlite3 = require("sqlite3").verbose();
const app = express();
const uuid = require("uuid");
const bodyParser = require("body-parser");
const cors = require("cors");

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

app.use(
  cors({
    origin: "*",
  })
);

const db = new sqlite3.Database("./db.db", sqlite3.OPEN_READWRITE, (err) => {
  if (err) {
    console.error(err.message);
    return;
  }
  console.log("Connected to SQLite database!");
});

app.use(express.json());

app.get("/", (req, res, next) => {
  res.json({ message: "Ok" });
});

// insert new users
app.post("/api/user/", (req, res, next) => {
  var errors = [];
  if (!req.body.password) {
    errors.push("No password specified");
  }
})
```

```

if (!req.body.email) {
  errors.push("No email specified");
}
if (errors.length) {
  res.status(400).json({ error: errors.join(",") });
  return;
}
var data = {
  username: req.body.username,
  email: req.body.email,
  password: req.body.password,
};
var sql = "INSERT INTO user (username, email, password) VALUES (?, ?, ?)";
var params = [data.username, data.email, data.password];
db.run(sql, params, function (err, result) {
  if (err) {
    res.status(400).json({ error: err.message });
    return;
  }
  res.json({
    message: "success",
    data: data,
    id: this.lastID,
  });
});
});

// user authentication (login page)
app.post("/api/login/", (req, res, next) => {
  var data = {
    username: req.body.username,
    password: req.body.password,
  };
  res.header("Access-Control-Allow-Origin", "*"); // Change this to your
actual origin if needed
  res.header(
    "Access-Control-Allow-Methods",
    "GET, POST, OPTIONS, PUT, DELETE, PATCH"
}

```

```

) ;
res.header("Access-Control-Allow-Headers", "Content-Type");

var errors = [] ;

if (!req.body.password) {
  errors.push("No password specified");
}
if (!req.body.username) {
  errors.push("No username specified");
}
if (errors.length) {
  res.status(400).json({ error: errors.join(",") });
  return;
}

var sql = `SELECT * FROM user WHERE username = ? AND password = ?`;
var params = [data.username, data.password];
db.run(sql, params, function (err, result) {
  if (err) {
    res.status(401).send({ message: "Unauthorized access" });
    return;
  }
  res.json({ message: " Authorized success" });
});
});

// timetable data
app.post("/api/timetable/", (req, res, next) => {
  const data = {
    major: req.body.major,
    year: req.body.year,
    semester: req.body.semester,
    day: req.body.day,
    time: req.body.time,
    courseName: req.body.courseName,
    roomNumber: req.body.roomNumber,
    courseType: req.body.courseType,
    instructorName: req.body.instructorName,
  }
});

```



```
};

res.header("Access-Control-Allow-Origin", "*"); // Change this to your
actual origin if needed

res.header(
  "Access-Control-Allow-Methods",
  "GET, POST, OPTIONS, PUT, DELETE, PATCH"
);

res.header("Access-Control-Allow-Headers", "Content-Type");

const sql = `INSERT INTO timetable (major, year, semester, time, day,
courseName, roomNumber, courseType, instructorName) VALUES (?, ?, ?, ?, ?, ?, ?,
?, ?, ?)`;

const params = [
  data.major,
  data.year,
  data.semester,
  data.time,
  data.day,
  data.courseName,
  data.roomNumber,
  data.courseType,
  data.instructorName,
];

db.run(sql, params, (err) => {
  if (err) {
    res.status(500).json({ error: "Error saving timetable" });
    return;
  }

  // Send session ID to client (e.g., in response body)
  res.json({ message: "Course Added Successfully!" });
});

// retrieve timetable
app.post("/api/showTimetable/", (req, res, next) => {
  const data = {
    major: req.body.major,
    year: req.body.year,
    semester: req.body.semester,
```



```
};

res.header("Access-Control-Allow-Origin", "*"); // Change this to your
actual origin if needed

res.header(
  "Access-Control-Allow-Methods",
  "GET, POST, OPTIONS, PUT, DELETE, PATCH"
);

res.header("Access-Control-Allow-Headers", "Content-Type");

const sql = `SELECT time, day, courseName, roomNumber, courseType,
instructorName FROM timetable WHERE major = ? AND year = ? AND semester = ?`;

const params = [data.major, data.year, data.semester];

db.all(sql, params, (err, rows) => {
  if (err) {
    res.status(500).json({ error: "Error fetching timetable" });
    return;
  }
  res.json({ timetable: rows });
});

}) ;

app.get("*", (req, res) => {
  res.send("404 Page Not Found");
});

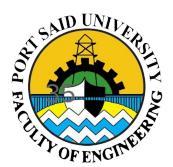
app.listen(3000, () => {
  console.log("Server listening on port 3000!");
});
```



## SQLite3

```
CREATE TABLE timetable (
    major TEXT,
    year TEXT,
    semester TEXT,
    time TEXT,
    day TEXT,
    courseName TEXT,
    roomNumber TEXT,
    courseType TEXT,
    instructorName TEXT,
    PRIMARY KEY (day, time, roomNumber)
);

CREATE TABLE user (
    ID INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT UNIQUE,
    email TEXT UNIQUE,
    password TEXT
);
```



## Challenges

**Securing the Gateway:** Our user authentication system struggles with granting access to unauthorized users.

**Taming the Edit Button:** The edit button offers unpredictable functionality for correcting course entries.

**Pinpointing the Right Timetable:** Retrieving the correct timetable based on a user's major, year, and semester proved challenging.

## Conclusion

Overall, the NHRR Timetable Generator project has the potential to significantly improve the organizing process of the timetable at Port Said University.

This student-centric application tackles the common challenges of scheduling conflicts and complex timelines by offering a personalized, visual, and readily adaptable solution.

Through its user-friendly interface, flexible options, and potential integration with university data, it can potentially lead to improved organization, efficiency, and academic success.

### Major Functional Improvements for the Timetable Generator

#### Integration with External Services and Platforms:

- Connect the timetable generator with university portals to automatically import course catalogs and updates.
- Allow synchronizing timetables with calendars and productivity tools for seamless cross-platform organization.

This enhances convenience and streamlines user workflow by integrating the generator into their existing ecosystem.

#### Data Visualization and Insights



- Offer visual representations of student schedules, like heatmaps or bar charts, highlighting workload distribution and potential issues.
- Generate reports and insights based on schedule data, analyzing course load, class frequency, and time management patterns.

This empowers users to gain a deeper understanding of their schedules and identify areas for improvement.

### AI-powered Course Advisor and Schedule Optimizer

Imagine a smart assistant within the timetable generator that understands your academic goals, learning style, and preferences. This AI assistant could offer personalized recommendations like:

- Course Recommendations: Suggesting courses based on your academic interests, past performance, and career aspirations, helping you discover hidden gems or avoid overloading yourself.
- Dynamic Schedule Optimization: Analyzing your existing schedule and suggesting adjustments to optimize workload distribution, minimize travel time, and maximize breaks for better focus.
- Personalized Study Planner: Automatically creating study plans based on your schedule, deadlines, and preferred learning methods, including reminders, resource suggestions, and progress tracking.
- Predictive Insights: Using data from the Port Said University community to predict workload intensity, professor difficulty levels, and upcoming deadlines, empowering you to prepare effectively.

■ ■ ■