The report includes my contribution in the project created by me and my colleagues. and my teamwork and dedication is reflected in this Project report.

Individual Project Report

Alex Johnson   ID: 104762702

Name:-   Alex Johnson

Student Id:-   104762702

Email:-   104762702@student.swin.edu.au

Date of Submission:- 03/11/2023

INDEX

Contents:-

1.Introduction

This report provides a comprehensive overview of my substantial contributions and the valuable learning outcomes derived from a collaborative project. The project revolved around a real world scenario and requriment of a summer school the project required a web application which can perform various tasks like login & attendance for students and Q&A Facilty for student and teacher and for teachers to select the time for class and for admin to create a course and soon this, offered me an excellent opportunity to explore various facets of software development and teamwork. Within this report, I will delve into my code contributions, specifically in JavaScript and CSS, highlight my design contributions, and discuss my influence on the overall team's performance.

2.Contributions

2.1 Code Contributions

My primary role in the project involved working on the JavaScript components, where I was responsible for ensuring the functionality of critical features. The following section provides a detailed account of my key code contributions and below I have provided the snippets of some the code I have developed and/or worked on and its explanation of how it works.

2.1.1 JavaScript Contributions:

I assumed a pivotal role in implementing essential functionalities, such as the creation of dynamic URLs for AJAX requests and the development of onclick functions that enabled the seamless integration of various features within this project. Below is an illustrative code snippet that showcases my JavaScript contributions:

JavaScript code

2.1.2 Code used on student side:

Studentinfo.html

```
<script th:inline="javascript">
    function chooseClass(id) {
        var url=/*[[ @{'/student/class/'} ]]*/;
        window.location.href=url+id;
    }
    function joinClass(id) {
        var url=/*[[ @{'/student/class/join/'} ]]*/;
        window.location.href=url+id;
    }
</script>
```

```
<script th:inline="javascript">
    var calendarInstance = new calendarJs( "calendar", {
        exportEventsEnabled: true,
    } );

    document.title += " v" + calendarInstance.getVersion();

    for (let studentClass of /*[[${studentClasses}]]*/ )
    {
```

```
        var event = {
            from: studentClass["teacherCourse"]["startTime"],
            to: studentClass["teacherCourse"]["endTime"],
            title: studentClass["course"]["courseName"],
            description: studentClass["course"]["description"]
        };

        calendarInstance.addEvent( event );
    }

</script>
```

In this code two JavaScript functions, **chooseClass** and **joinClass**, are defined. These functions take an **id** as a parameter and construct URLs based on that **id** using Thymeleaf expressions. The constructed URLs are then used to navigate to different class-related endpoints when invoked, allowing a student to choose or join a class based on the provided **id**.

In the second part using Thymeleaf and JavaScript,I have initialized a calendar with the id "calendar" using the **calendarJs** library. It enables event exporting and appends the calendar version to the document title. Then, it iterates through a collection of student classes (likely provided by the Thymeleaf variable **${studentClasses}**) and creates calendar events for each class. The events are defined with start and end times, a title extracted from the course information, and a description. These events are added to the calendar instance, populating the calendar with class schedule information.

Fig 1.4 Student Enrolled Course

Students can choose the course they want to enroll in .

Students can choose the course they want to enrol and the class according to their preferences

Current class(Schedule)

Classroom.html

```javascript
<script th:inline="javascript">
    function question() {
        var url = /*[[ @{'/student/message/save'} ]]*/;
        $.ajax({
            type: "POST",
            url: url ,
            data: $('#message').serialize(),
            success: function (data) {
                $("#refresh").html(data);
            },
            error:function () {
                alert("error");
            }
        });
    }

    $(document).ready(function() {
        var classId = $("#teacherCourseId").val();
        var url = /*[[ @{'/message/refresh/'} ]]*/;
        setInterval(function() {
            $.get(url + classId, function(data) {
                $('#refresh').html(data);
            });
        }, 20000);
    });

    $(document).ready(function() {
```

```
        var classId = $("#teacherCourseId").val();
        var url = /*[[ @{'/message/attend/'} ]]*/;
        setInterval(function() {
            $.get(url + classId, function(data) {
                $('#attend').html(data);
            });
        }, 20000);
    });
</script>
```

This code is used in the chat box which allows user to ask questions in this code using Thymeleaf and JavaScript, there are three functions. The `question` function sends a POST request to the "/student/message/save" endpoint with data serialized from the "message" form. It updates the "refresh" element with the response data on success and displays an alert on error.

Two other functions are triggered when the document is ready. They both use `setInterval` to periodically send GET requests to different endpoints ("/message/refresh/" and "/message/attend/") with a class ID parameter obtained from an element with the ID "teacherCourseId." The response data is used to update the "refresh" and "attend" elements, respectively. These functions refresh the content of the specified elements every 20 seconds, likely for real-time updates in a messaging or attendance system within a web application.
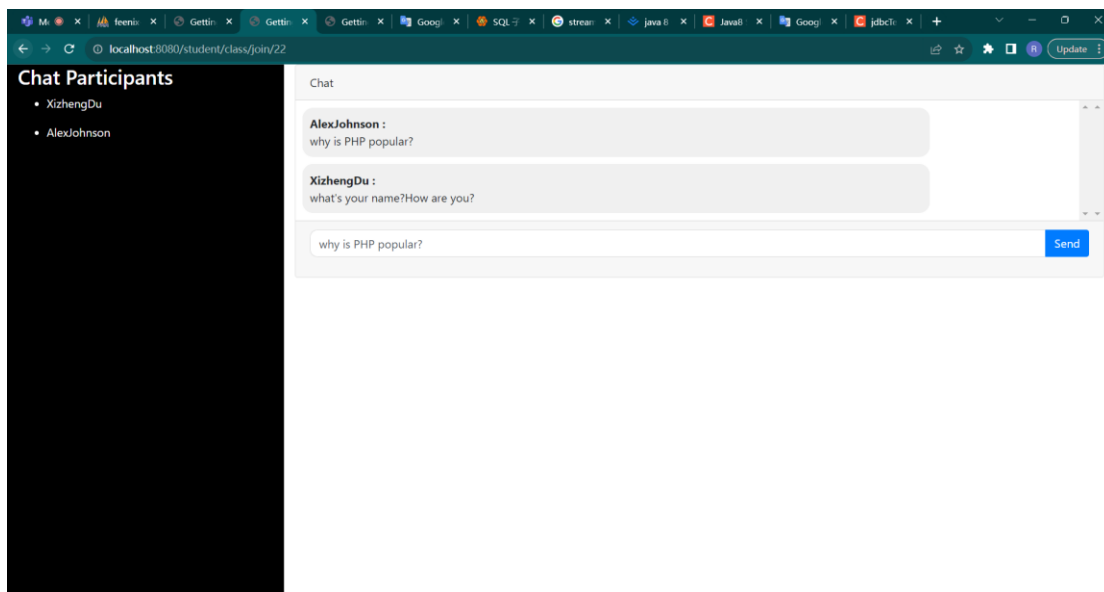
Code for teacher side:

Teacherinfo.html

```
<script th:inline="javascript">
    function joinClass(id) {
        var url=/*[[ @{'/teacher/class/join/'} ]]*/;
        window.location.href=url+id;
    }
</script>
```

In this code using Thymeleaf and JavaScript, there is a single function named **joinClass**. This function takes an **id** as a parameter and uses Thymeleaf expressions to construct a URL for joining a teacher's class based on the provided **id**. When invoked, it redirects the user's browser to the constructed URL, enabling them to join the specified class.

```javascript
<script th:inline="javascript">
    var calendarInstance = new calendarJs( "calendar", {
        exportEventsEnabled: true,
    } );

    document.title += " v" + calendarInstance.getVersion();

    for (let teacherCourse of /*[[${teacherCourses}]]*/ )
    {
        var event = {
            from: teacherCourse["startTime"],
            to: teacherCourse["endTime"],
            title: teacherCourse["course"]["courseName"],
            description: "A description of the new event"
        };

        calendarInstance.addEvent( event );
    }

</script>
```

This code is similar to that of the studentinfopage which allows student to view there upcoming registerd classes well in this case it show the teachers the classes they have created.

teacherClassroom.html
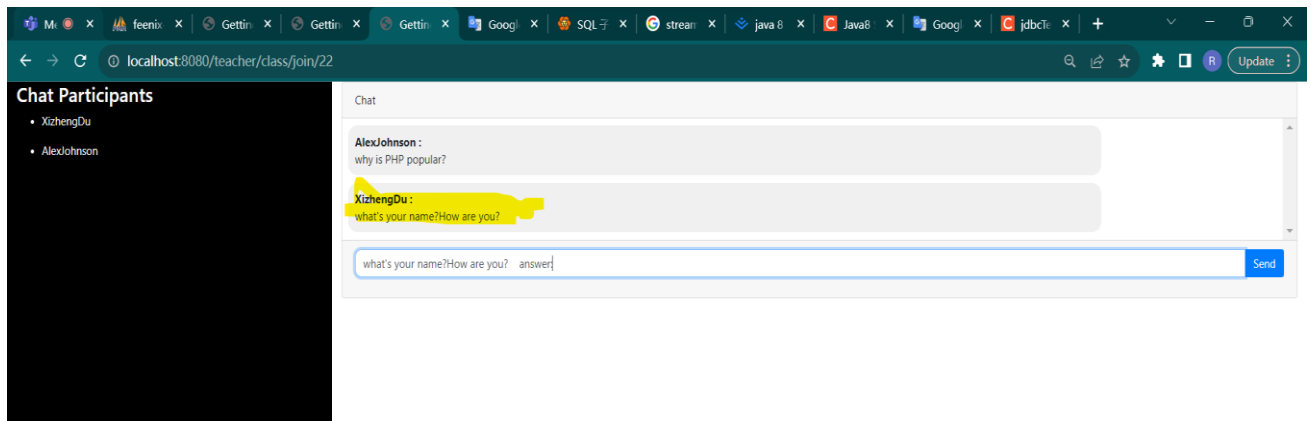
```html
<script th:inline="javascript">
    function respondToQuestion(messageId, studentId, questionId) {
        var messageTestId = '#message'+messageId;
        $("#message-input").val($(messageTestId).text()+'    answer:');
        $("#studentId").val(studentId);
        $("#questionId").val(questionId);
    }

    function question() {
        var url = /*[[ @{'/teacher/message/save'} ]]*/;
        $.ajax({
            type: "POST",
            url: url ,
            data: $('#message').serialize(),
            success: function (data) {
                $("#refresh").html(data);
            },
            error:function () {
                alert("error");
            }
        });
    }

            $(document).ready(function() {
        var classId = $("#teacherCourseId").val();
        var url = /*[[ @{'/message/refresh/'} ]]*/;
        setInterval(function() {
            $.get(url + classId, function(data) {
                $('#refresh').html(data);
            });
        }, 20000);
    });

    $(document).ready(function() {
        var classId = $("#teacherCourseId").val();
        var url = /*[[ @{'/message/attend/'} ]]*/;
        setInterval(function() {
            $.get(url + classId, function(data) {
                $('#attend').html(data);
            });
        }, 20000);
    });
</script>
```

This code snippet is similar to that of classroom.html which is used in student side which allows the students to post questions with only on edtion of respondtoquestion function which allows the teacher to respond to a specific questions.

 Answering Questions

Admin side
CourseList.html

```javascript
<script th:inline="javascript">
var xhr = false;
if (window.XMLHttpRequest)
{
xhr = new XMLHttpRequest();
}
else if (window.ActiveXObject)
{
xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
function deleteCourse(id) {
  var txt;
  if (confirm("Are you sure to delete this course?")) {
    xhr.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            location.reload();
        }
    };
    var url=/*[[ @{'/course/delete/'} ]]*/;
    xhr.open("get", url+id, true);
    xhr.send();
  }
}
</script>
```

In this admin side side code using Thymeleaf and JavaScript, an XMLHttpRequest (XHR) object, `xhr`, is created to facilitate AJAX (asynchronous) communication. The script first checks for compatibility with modern browsers and legacy Internet Explorer.

The `deleteCourse` function is defined to handle course deletion. When invoked, it shows a confirmation dialog to ensure the user wants to delete the course. If confirmed, it sets up an event handler for the XHR object to reload the page upon a successful response (HTTP status 200). It then constructs a URL for the course deletion endpoint based on the provided `id`, opens a GET request with the URL, and sends the request asynchronously to delete the course. If the user confirms, the page is reloaded after the course is deleted. This script appears to be used for deleting courses in a web application with user confirmation.
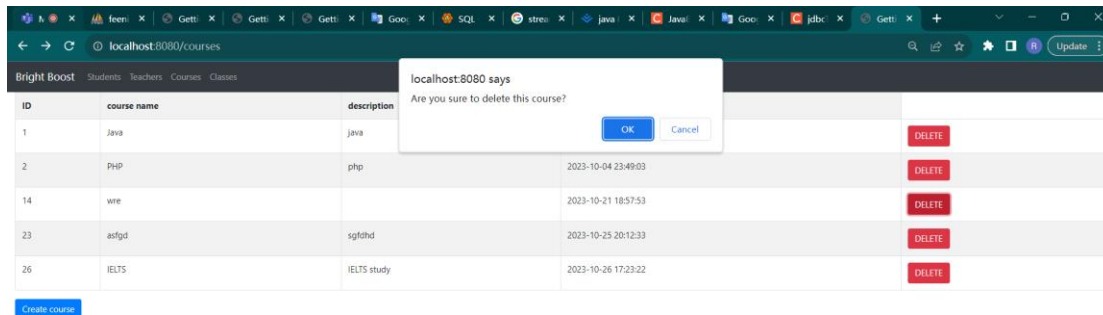
This codes allowed us to [explain the specific functionality] and greatly improved the user experience.
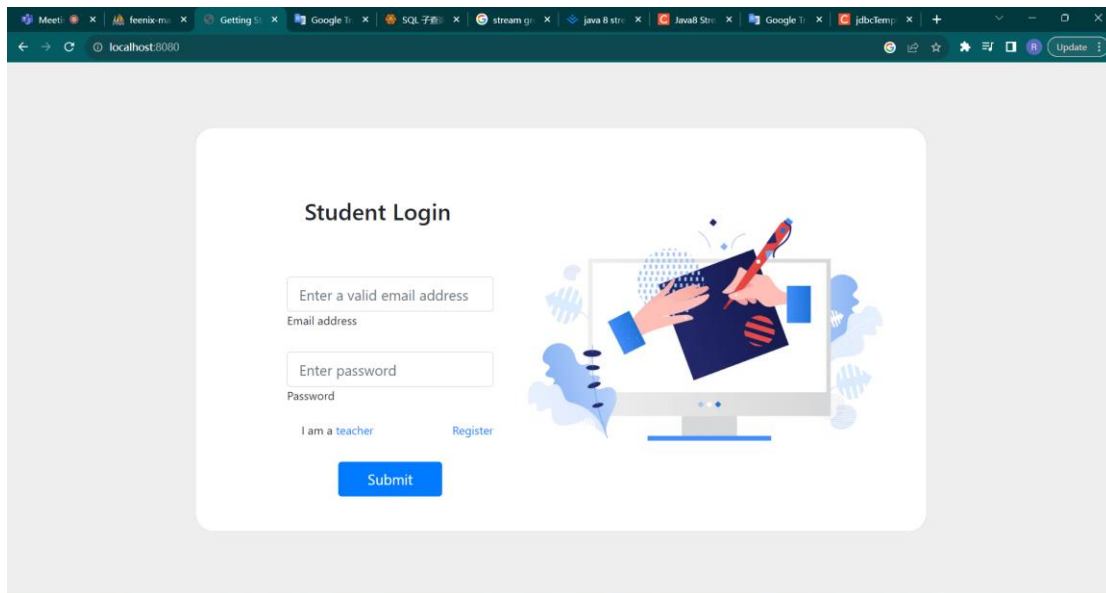
2.1.3 CSS Contributions:

In addition to JavaScript, I also worked on the project's CSS, refining the visual aesthetics and layout.

In this visual side I provide my creative expertise in this area provide simple and usable classes of bootstrap like this used in

 Studentlogin.html

```
<section class="vh-100" style="background-color: #eee;">
    <div class="container h-100">
        <div class="row d-flex justify-content-center align-items-center h-
100">
            <div class="col-lg-12 col-xl-11">
                <div class="card text-black" style="border-radius: 25px;">
                    <div class="card-body p-md-5">
                        <div class="row justify-content-center">
                            <div class="col-md-10 col-lg-6 col-xl-7 d-flex
align-items-center order-1 order-lg-2">
```

These CSS enhancements had a profound impact on the project's visual appeal and usability.

2.2 Design Contributions

My involvement extended beyond just coding; I also made significant contributions to the project's design and architecture.

2.2.1 User Interface Design:

I actively participated in the user interface design process, proposing and implementing improvements. Here are some visual enhancements I introduced:

I convert the pages from this to this
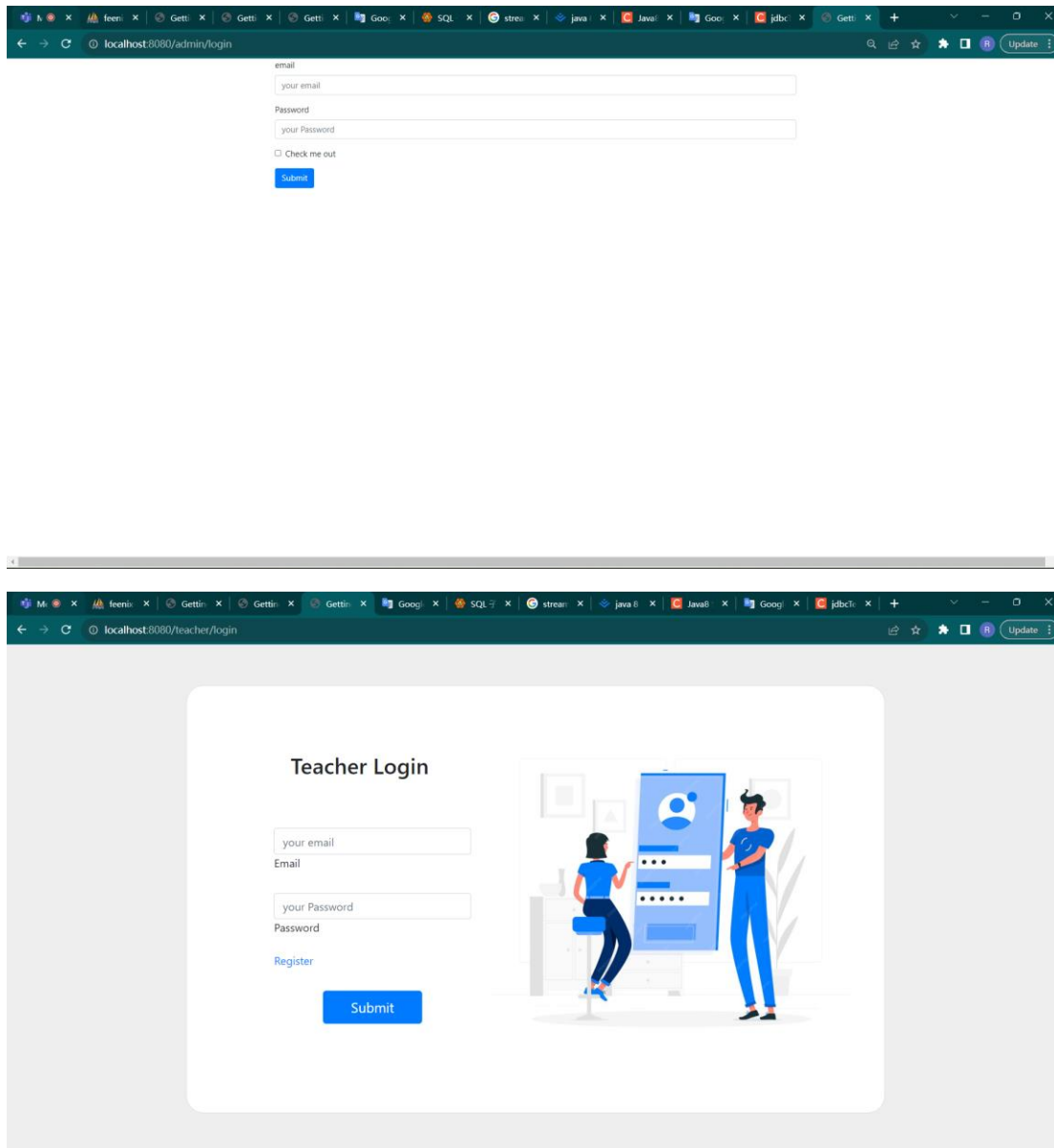
These changes made the user interface more intuitive and visually appealing, resulting in a more engaging user experience.

2.2.2 Software Architecture:

Collaborating in the team, I provided valuable insights into the project's software architecture. I Provided my insight towards developing the workflow of the project, which included the creation of essential architectural diagrams.

2.2.3 Activity Diagram:

One of the key contributions in the software architecture was the creation of an activity diagram that depicted the workflow of the project. This diagram illustrated the sequential steps and interactions within the system, highlighting the various stages and decision points in our application. Like that of the student and how it worked and what functionality were given to a student and then similarly to teacher and admin.

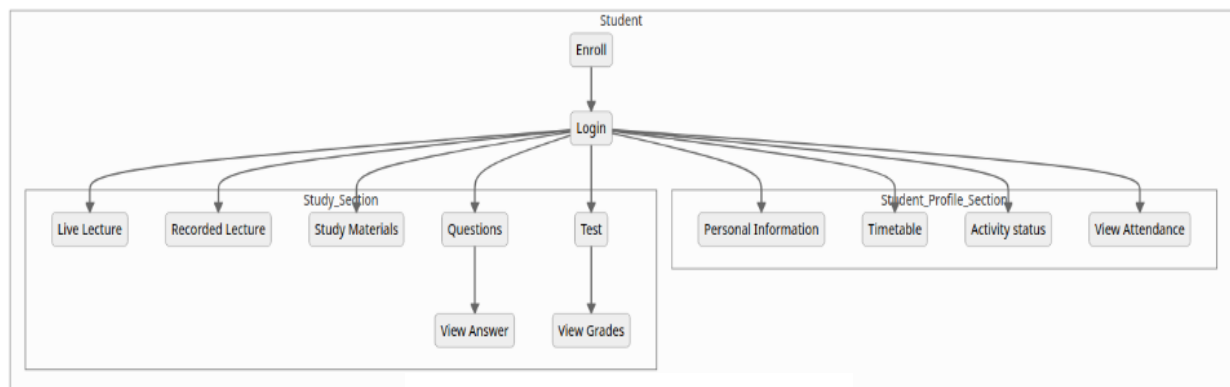Visualizing the workflow of key processes within the portal.
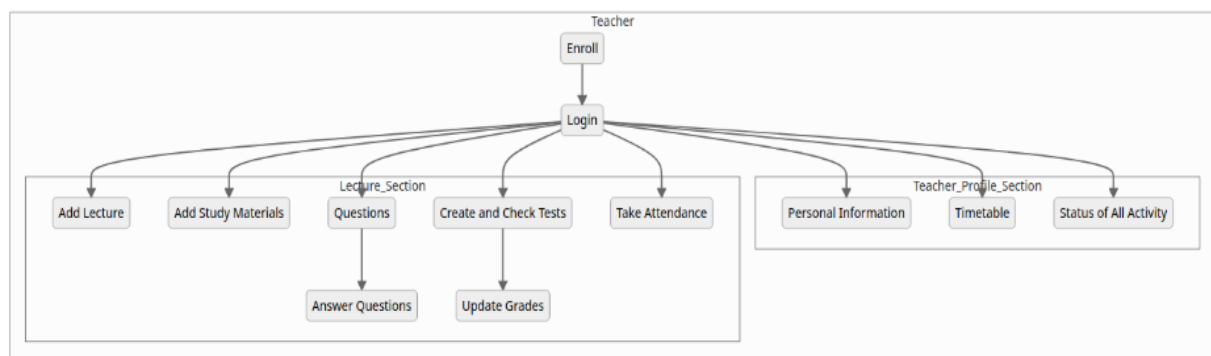

Fig 4.2.1 Activity Diagram for Student

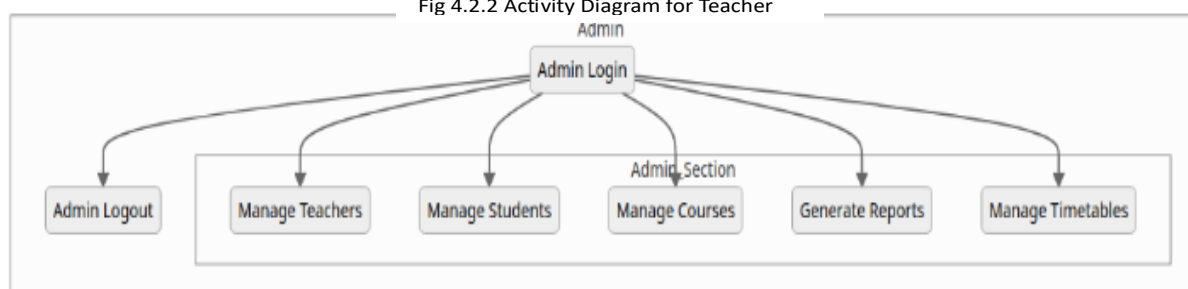Fig 4.2.2 Activity Diagram for Teacher

Fig 4.2.3 Activity Diagram for Admin

The activity diagram not only served as a valuable reference for our team but also clarified the flow of information and user interactions. It helped us identify potential bottlenecks and optimize the process for enhanced efficiency.

2.2.4 Entity-Relationship (ER) Diagram:

Another significant architectural contribution was the development of an Entity-Relationship (ER) diagram. This diagram captured the essential entities, their attributes, and the relationships between them in our database schema.

Depicting the database structure and relationships among different entities.
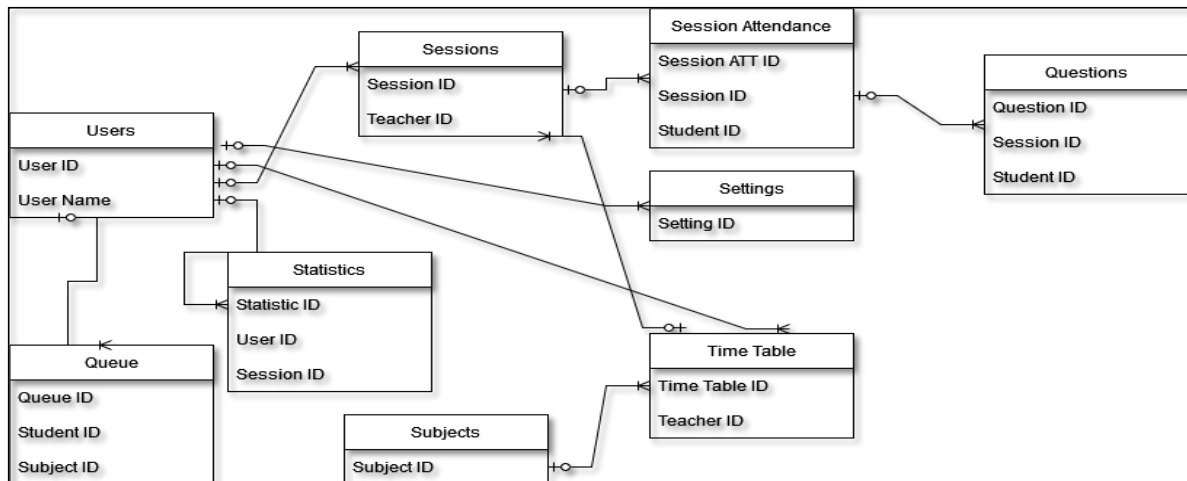
The ER diagram played a vital role in shaping the database design. It provided a clear visual representation of the data structure and relationships, ensuring that our database schema was well-organized and efficient.

These architectural changes had a profound impact on the project's overall structure, enhancing its scalability and maintainability. The activity diagram streamlined the project workflow, while the ER diagram ensured a robust and efficient database structure. These contributions were instrumental in achieving a well-structured and adaptable system.

2.3 Team Contributions

I made a consistent effort to be a proactive, trustworthy team player who was totally dedicated to achieving our common objectives throughout the project. My commitment to our team's performance was demonstrated in a number of ways.

In terms of leadership responsibilities, I took the lead at pivotal points in the project and led the group through significant choices and difficulties. The project gained momentum and eventually succeeded because of my ability to give direction and stay focused.

Another area I had a big influence on was team engagement. I took an active part in brainstorming sessions, meetings, and group projects, creating an atmosphere where we could all explore different viewpoints and come up with creative ideas. This involvement improved our project and inspired the group to collaborate well.

One of my main contributions was finishing the work on time. I constantly completed my work ahead of schedule, preventing possible bottlenecks in the project and promoting a smooth workflow. This consistent performance increased our work's overall efficiency while also lowering team stress.

In addition, I was crucial to the organization of the team. I made sure that everything was planned out and executed flawlessly, efficiently allocating tasks, scheduling, and allocating resources. This methodical approach made it easier for us to stay on task and produce at our highest level.

I continued to participate actively in our team's activities throughout the project in addition to these significant contributions. I actively participated in conversations, showed up to all scheduled meetings, and regularly offered criticism and recommendations to better the course of our project. My dedication to efficient communication and teamwork was further demonstrated by the frequent email and chat exchanges I had with team members, which guaranteed openness in our work.


30 August

30-08 18:17

To create a database for the software application, you would need to design several tables to store different types of data related to teachers, students, sessions, questions, and more. Here are some of the required tables for the described application:

1. **Users Table:**
   - Stores information about teachers, students, and administrators.
   - Fields: UserID (Primary Key), Username, Password, Role (Teacher/Student/Admin), First Name, Last Name, Email, etc.
2. **Sessions Table:**
   - Stores information about tutoring sessions conducted by teachers.
   - Fields: SessionID (Primary Key), TeacherID (Foreign Key), Date and Time, Duration, Subject/Area, Location, etc.
3. **SessionAttendance Table:**
   - Tracks which students attended each session.
   - Fields: SessionAttendanceID (Primary Key), SessionID (Foreign Key), StudentID (Foreign Key), AttendanceStatus (Attended/Absent), etc.
4. **Questions Table:**

Stores information about tutoring sessions conducted by teachers.
   - Fields: SessionID (Primary Key), TeacherID (Foreign Key), Date and Time, Duration, Subject/Area, Location, etc.
3. **SessionAttendance Table:**
   - Tracks which students attended each session.
   - Fields: SessionAttendanceID (Primary Key), SessionID (Foreign Key), StudentID (Foreign Key), AttendanceStatus (Attended/Absent), etc.
4. **Questions Table:**
   - Stores information about individual questions asked during sessions.
   - Fields: QuestionID (Primary Key), SessionID (Foreign Key), StudentID (Foreign Key), QuestionContent, TimeAsked, TimeTaken, Answered (Yes/No), etc.
5. **Subjects Table:**
   - Lists the available subject areas.
   - Fields: SubjectID (Primary Key), SubjectName, Description, etc.
6. **Timetable Table:**
   - Defines the schedule of sessions for each teacher.
   - Fields: TimetableID (Primary Key), TeacherID (Foreign Key), Day, TimeSlot, SubjectID (Foreign Key), etc.
7. **Statistics Table:**
   - Stores calculated statistics and summaries for sessions and user activities.
   - Fields: StatisticID (Primary Key), UserID (Foreign Key), SessionID (Foreign Key), QuestionsAnswered, AttendanceRate, etc.
8. **Queue Table:**
   - Manages student requests for assistance.
   - Fields: QueueID (Primary Key), StudentID (Foreign Key), SubjectID (Foreign Key), RequestTime, Status (Pending/Answered), etc.
9. **Settings Table:**
   - Stores application settings and configurations.
   - Fields: SettingID (Primary Key), SettingName, SettingValue, Description, etc.

Each table represents a specific type of data that the application needs to manage. Foreign keys establish relationships between tables. You may need additional tables based on the specific features and functionalities you want to include in the application. Always ensure that your database design follows good practices for data integrity, normalization, and security.
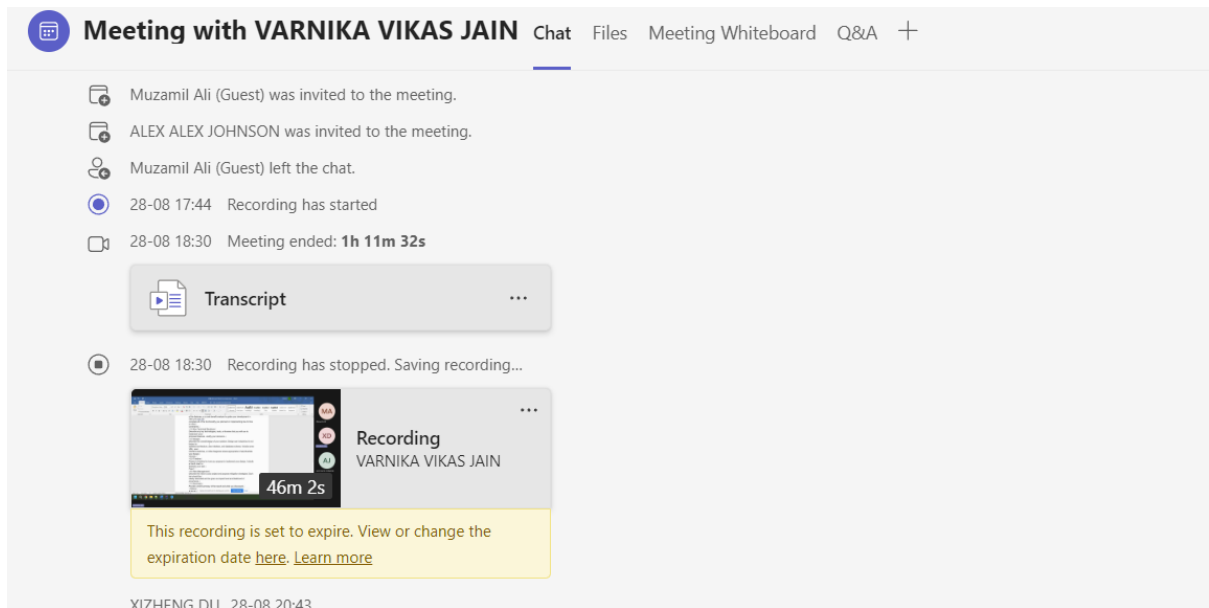
A one of my contribution towards the project while deciding the table for the Project
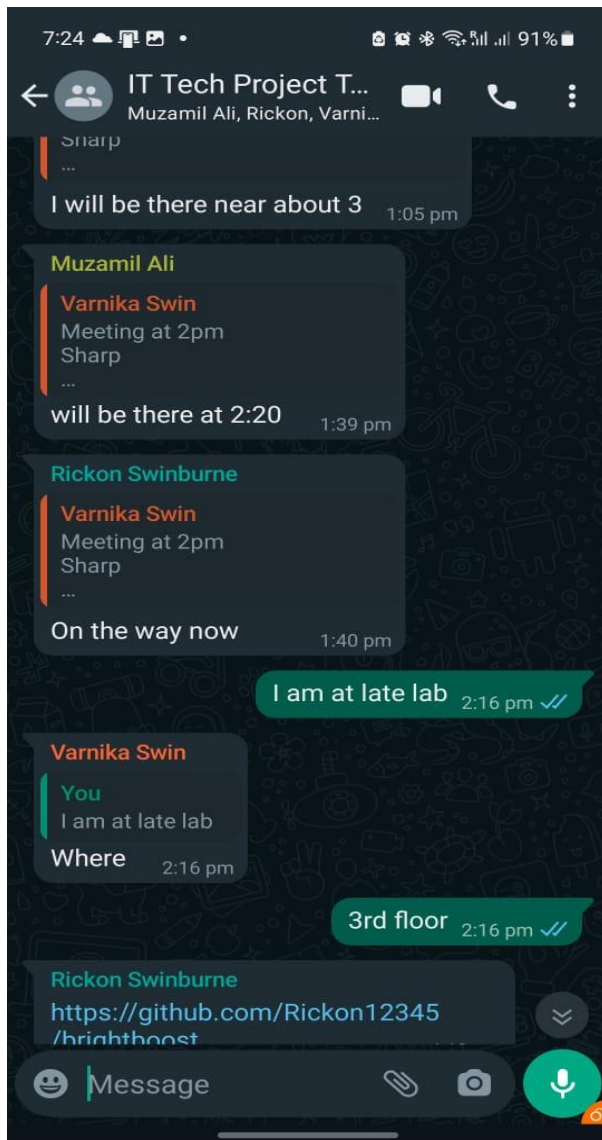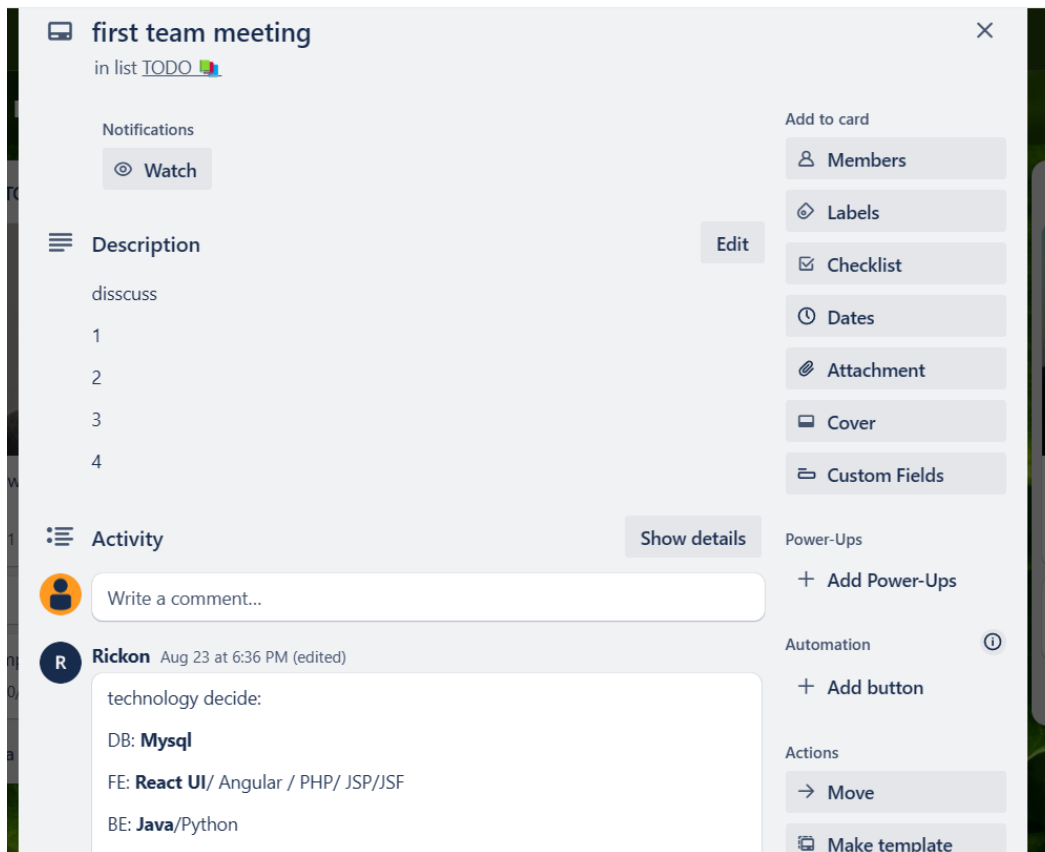
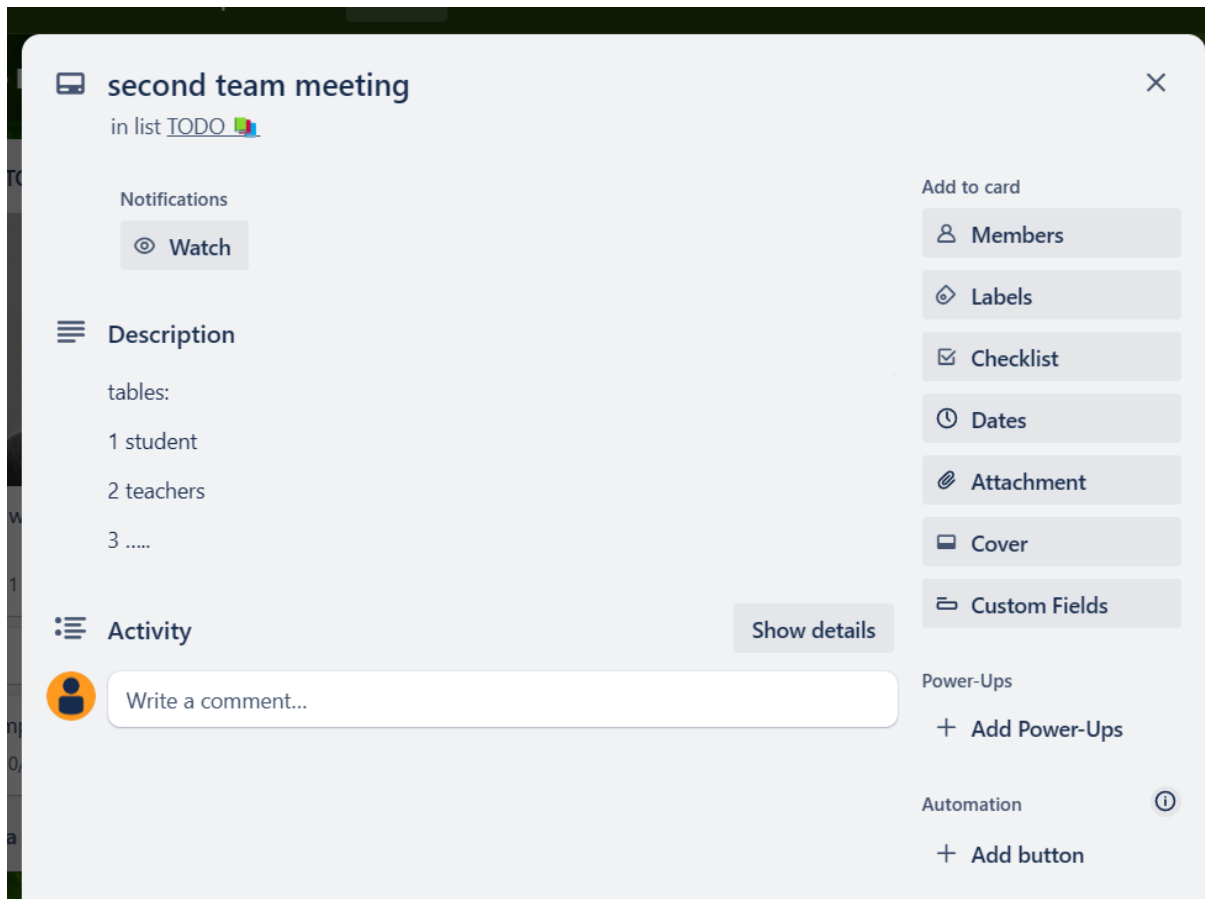Screeenshot of one of our discussion before grp meeting.
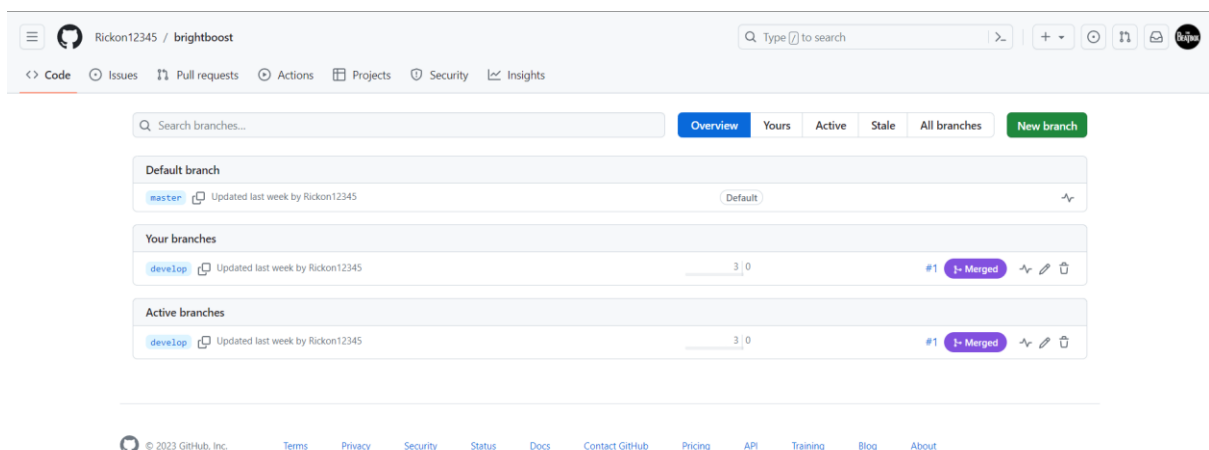
Screenshot of github we used to merge and create the project, showing my contribution aswell..

Screenshot of one of the online meeting we did using teams

Few of Screenshot of using Trello to keep track of the meetings

In summary, my multifaceted contributions and dedication to effective teamwork and collaboration significantly enhanced our team's performance and project outcomes. The successful completion of our project was the result of a collective effort, and I take great pride in the role I played in our achievements.

3. Learning Outcome Reflection

In this comprehensive section, I will reflect on how I effectively met each of the unit learning outcomes and how this collaborative project significantly contributed to my substantial personal and professional growth. This transformative experience allowed me to exceed the expectations of these learning outcomes, equipping me with a robust skill set and valuable insights that will undoubtedly serve me well in my future endeavour's.

3.1 Learning Outcome 1: Understanding of Programming

My comprehension of programming has been greatly improved by my participation in this project. I was able to apply difficult programming concepts to real-world problems through practical experience, which helped me learn more and improve my problem-solving abilities. I went deep into the nuances of various programming languages and their real-world uses on this never-ending learning journey. This experience gave me more technical proficiency and gave me the confidence to take on challenging programming tasks in the future.

3.2 Learning Outcome 2: Reflective Practice and Self/Peer Evaluation

A key factor in the success of our project was my dedication to reflective practice and regular assessment of my own and my peers' contributions. Our work was continuously improved as a result of this iterative process. We produced better project outcomes by identifying areas for improvement and making the required adjustments after critically evaluating our performance. My personal and professional development has also benefited greatly from this reflective approach, which has taught me the value of self-awareness, flexibility, and ongoing improvement.

3.3 Learning Outcome 3: Applying Knowledge of Programming Fundamentals

I used my foundational understanding of programming to tackle the particular technological issues our project brought up. My comprehension of these basic ideas was reinforced by this hands-on experience, which also demonstrated how applicable they are in everyday life. It was a life-changing experience that taught me how to effectively adapt and apply programming concepts to solve real-world issues. My capacity for original thought and innovation in the field of programming has improved thanks to this link between theory and practice.

3.4 Learning Outcome 4: Learning Collaboratively in a Team Environment

Our project's collaborative nature served to highlight how crucial good collaboration is. I gained experience working well with a variety of team members and fostering a positive team environment. My perspective has been widened by this experience, and I now recognize the value of the many abilities and perspectives that team members have to offer. It also underlined how important compromise, open communication, and shared accountability are to reaching shared objectives. This learning objective has had a long-lasting effect on my capacity for cooperative teamwork.

3.5 Learning Outcome 5: Finding, Organizing, and Evaluating Information

I conducted a lot of study for the project to solve a variety of technological issues. My ability to locate trustworthy sources, arrange information effectively, and make well-informed judgments based on research has improved because of this experience. These abilities are beneficial for problem-solving in many fields and for lifelong learning in addition to project work. Information gathering, analysis, and synthesis are essential skills that will help me in all of my future endeavour's.

3.6 Learning Outcome 6: Awareness of Social and Cultural Perspectives

Our team's diversity and the collaborative nature of the project underscored the significance of social and cultural perspectives in project development. This awareness enhanced our collective decision-making process as we actively considered different viewpoints and cultural sensitivities. It reinforced the importance of inclusivity, empathy, and cultural awareness in the development of technology solutions. This experience broadened my understanding of how technology interacts with various social and cultural contexts, emphasizing the need for ethical and culturally sensitive approaches in the ever-evolving tech industry.

In summary, the project provided a comprehensive and transformative learning experience. I firmly believe that I successfully met each of the unit learning outcomes, and this experience has not only expanded my technical knowledge but also enriched my interpersonal and critical thinking skills. I

feel well-prepared for future challenges and opportunities in the field of programming and beyond, as this experience has shaped me into a more effective, adaptable, and knowledgeable professional.

4. Summary

in summary, my contributions to this project were multifaceted and had a significant impact across various aspects of our work. I was a major contributor to the design of the JavaScript and CSS elements, which ensured their functionality and improved their appearance. This coding knowledge was crucial to producing a product that was both functional and aesthetically beautiful.

In addition, my contribution to the design and architecture aspects was crucial in enhancing the project's overall efficiency and structure. I contributed to making the project's development process more efficient and productive by providing design insights. This teamwork was crucial to achieving the project's goals and ensuring that it was carried out precisely.

Throughout the team, I played a leadership role by helping the group through difficult choices and obstacles. These acts of leadership helped keep the project on track and moving forward, helping us achieve our objectives successfully. Furthermore, my regular participation in brainstorming sessions, team meetings, and cooperative projects promoted an atmosphere of candid communication and creativity. I gave insightful comments and ideas during conversations, fully participating in the process and helping to lead our project in the proper path.

My contributions were characterized by my punctuality in finishing duties, as I often turned in my work ahead of schedule. This lowered the possibility of project bottlenecks and made it easier for the team as a whole to work together smoothly.My commitment to timeliness was reflective of my dedication to the project's success.

I was essential in keeping the team organized by efficiently allocating duties, scheduling, and allocating resources. This made sure that our group approached the job in an organized manner, encouraging productivity and making the most use of our available resources. My dedication to organization made it possible for our project to stay on course and on schedule.

My contributions overall covered a broad spectrum of interpersonal and technical abilities, demonstrating my commitment to productive cooperation and the project's success. We accomplished our goals as a team, and I'm proud of my part in what we accomplished as a whole.

References

- MozDevNet, M.D.N. (no date a) *CSS: Cascading style sheets*, *MDN*. Available at: https://developer.mozilla.org/en-US/docs/Web/CSS (Accessed: 27 October 2023).
- MozDevNet, M.D.N. (no date c) *JavaScript*, *MDN*. Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript (Accessed: 27 October 2023).
- W3Schools. (no date) *Learn to code*, *W3Schools Online Web Tutorials*. Available at: https://www.w3schools.com/ (Accessed: 27 October 2023).
- Dauer, J. *et al.* (2023) *A list apart*, *A List Apart*. Available at: https://alistapart.com/ (Accessed: 27 October 2023).
- Webdesigner Depot (no date) *Web design blog*, *WDD*. Available at: https://www.webdesignerdepot.com/ (Accessed: 27 October 2023).
- GitHub (no date) GitHub Pages. Available at: https://pages.github.com/ (Accessed: 03 November 2023).
- Stackoverflow (no date) Where developers learn, share, & build careers, Stack Overflow. Available at: https://stackoverflow.com/ (Accessed: 03 November 2023).