

Attendance Website Management Technical Document

1. วัตถุประสงค์ของโปรแกรม

เว็บไซต์นี้สร้างขึ้นเพื่อจัดการการเช็คอินอย่างเป็นระบบและสามารถดูข้อมูลและจัดการรายละเอียดต่าง ๆ ได้อย่างชัดเจน โดยครอบคลุมข้อมูลในส่วนของข้อมูลรายวิชา ข้อมูลอาจารย์ และข้อมูลนิสิต นอกจากนี้ยังมีระบบสำหรับการส่งใบลาขาดสำหรับนิสิตได้

ทั้งหมดเพื่อให้การเช็คอินเข้าชั้นเรียนในแต่ละครั้งสามารถดำเนินไปได้อย่างสะดวกและรวดเร็ว

<https://github.com/toptanawit/attendace-website>

<https://www.figma.com/file/tj39ZKMgyLUwwOEdae6DkF/attendance-web-project?node-id=0%3A1&t=d28YxoqVfs2EDK2q-0>

2. ระบบของโปรแกรม

ผู้ใช้งานจะแบ่งออกเป็น 3 กลุ่ม คือ admin อาจารย์ และนิสิต โดยจะมีความสามรถต่างกัน ดังนี้

Admin

- สามารถจัดการข้อมูลใน database โดย CRUD ได้ทุกส่วน ตั้งแต่ข้อมูลของนิสิต ข้อมูลอาจารย์ และข้อมูลรายวิชา โดยสามารถเพิ่ม ลบ หรือแก้ไขได้ทุกส่วน เช่น เพิ่มรายวิชาและนิสิตที่ลงทะเบียนเรียนในรายวิชา ลบข้อมูลนิสิต เป็นต้น

อาจารย์

- สามารถดูรายวิชาที่สอนได้อย่างละเอียด สามารถดูได้ว่านิสิตคนใดมาเข้าเรียนกี่ครั้ง แต่จะไม่สามารถดูรายวิชาอื่นได้
- สามารถดูใบงานที่นิสิตส่งมาได้ และสามารถแก้ไขสถานะของใบงานได้
- สามารถสร้างรหัสที่ใช้สำหรับการเช็คอินในแต่ละคาบได้

นิสิต

- สามารถดูข้อมูลรายวิชาที่ลงทะเบียนเรียนได้อย่างละเอียด
สามารถดูจำนวนการเข้าเรียนและรายละเอียดการเข้าเรียนแต่ละครั้งของตนเองได้

- สามารถส่งใบลาในกรณีลาป่วยหรือลากิจให้กับอาจารย์ประจำวิชาได้
- สามารถใช้ระบบเช็คชื่อได้ โดยจะกรอกข้อมูลรายวิชาที่เข้าเรียนและรหัสของการเช็คชื่อในแต่ละครั้ง

3. วิธีเริ่มต้นการใช้งาน

เบื้องต้นจะต้องเริ่มจากการเพิ่ม user ที่เป็น admin เข้าไปใน database ก่อน โดยให้ใส่ username ตามต้องการและ status เป็น admin แต่เนื่องจาก password นั้นจำเป็นต้องผ่านการ hash ก่อน จึงต้องไปที่ sign up ก่อนแล้วจึงกรอก username ของ admin ที่เพิ่มเข้าไป และตั้ง password ตามต้องการที่ส่วนนี้ จากนั้นจะสามารถเข้าสู่ระบบเพื่อเริ่มใช้งานได้ โดย admin จะสามารถเพิ่ม user ทั้งของนิสิตและของอาจารย์ได้ รวมถึงเพิ่มรายวิชาและนิสิตที่ลงเรียนในรายวิชาได้ เพื่อให้นิสิตและอาจารย์นำข้อมูลไปใช้ในส่วนของตัวเองต่อไป

โดย database ที่มีการแบบให้คู่กับ project นี้จะมีการเพิ่ม user โดยมี username คือ admin, co631010318, co631010325, co631010382 และ password ของทุก user จะเป็น 1234 เพื่อการทดสอบ รวมถึงมีการเพิ่มรายวิชา CSC361 Web Programming และการเช็คชื่อของรายวิชาด้วย นอกจากนี้จะมีระบบการยืนใบลาที่ได้เพิ่มข้อมูลเบื้องต้นไว้ เช่นเดียวกัน

4. Development Technologies

Frontend

ในส่วนของ frontend จะใช้ EJS และ HTML เป็นหลัก และใช้ framework หรือดึง api ด้านนอกมาใช้ในบางส่วน

Language

- HTML: ใช้สำหรับจัดการข้อมูลที่ใช้ในการแสดงผล
- CSS: ใช้สำหรับการจัดการ User Interface
- JavaScript: ใช้สำหรับจัดการการทำงานของ Element ต่างๆ รวมถึงส่งข้อมูลไปยัง Backend
- EJS: ใช้สำหรับดึงข้อมูลจาก Backend นำมาแสดงผล

Framework

- Bootstrap: ใช้สำหรับจัดการเกี่ยวกับ User Interface โดยในโปรเจคนี้ใช้ร่วมกับ CSS

Library

- Jquery: ใช้สำหรับจัดการในส่วนการทำงานของ Element ต่างๆ

API (ภายนอก)

- QRServer: ใช้สำหรับอ่าน QR Code
- IPApi: ใช้สำหรับตรวจสอบ IP Address ของผู้ใช้งาน
- Google Chart: ใช้สำหรับสร้าง QR Code

ตัวอย่าง Code บางส่วน

```
<!-- Table -->
<section>
  <div class="mb-3 d-flex justify-content-between">
    <input type="text" id="myInput" name="myInput" placeholder="Search for students" class="form-control w-100 me-4">

    <a href="/add-students"><button type="button" class="btn text-white"
      style="width: 160px; background-color: #007CB7;">+ Add student</button></a>
  </div>
  <div class="table-responsive" id="no-more-tables">
    <table id="myTable" class="table bg-white table-striped">
      <thead class="text-dark">
        <tr>
          <th>Student ID</th>
          <th>Name</th>
          <th>Major</th>
          <th>Email</th>
          <th>Phone</th>
        </tr>
      </thead>
      <tbody id="tableData">
        <% for(var i=0; i < students.length; i++) { %>
          <tr>
            <td><%= students[i].student_id %></td>
            <td><%= students[i].name %></td>
            <td><%= students[i].major %></td>
            <td><%= students[i].email %></td>
            <td><%= students[i].phone %></td>
            <td><a href="/students/<%= students[i].buasri_id %>"><i class="bi bi-chevron-right fs-4"></i></a></td>
          <% } %>
        </tr>
      </tbody>
    </table>
  </div>
</section>
<!-- Table End-->
```

ส่วนนี้จะเป็นหน้าที่แสดงผลนิสิตทั้งหมด โดยจะมีการใช้ ejs เพื่อรับข้อมูลจาก backend มาแสดงผล

โดยข้อมูลที่ได้นั้นจะเป็น array จึงต้องนำมา for loop เพื่อสร้าง table ให้มี row

ตามจำนวนของข้อมูลทั้งหมด

```
<!-- Detail Course Start -->


<form method="POST" action="/add-students">
<div class="my-4 d-flex justify-content-between align-items-center">
    <h1>Students > Add student</h1>
    <div class="d-flex justify-content-between">
        <a href="/students" class="h-50">
            <button type="button" class="btn text-white me-3"
                style="width: 100px; background-color: #C5CED1;">Cancel</button>
        </a>
        <button type="submit" class="btn text-white h-50"
            style="width: 100px; background-color: #007CB7;">Submit</button>
    </div>
</div>

<div class="row">
<div class="col-12 col-sm-6 justify-content-md-ceneter">
    <div class="mb-3">
        <label for="student_id" class="form-label fw-bold">Student ID</label>
        <input type="text" class="form-control" name="student_id" id="student_id" aria-describedby="student_id"
            placeholder="Student ID" required>
    </div>
</div>


```

```

<div class="col-12 col-sm-6 justify-content-ceneter">
    <div class="mb-3">
        <label for="major" class="form-label fw-bold">Major</label>
        <select type="major" class="form-select" name="major" id="major" required>
            <option disabled selected>Open this select your major</option>
            <option value="Interactive and Multimedia Design">Interactive and Multimedia Design</option>
            <option value="Cyber Business Management">Cyber Business Management</option>
            <option value="Computer Innovation for Communication">Computer Innovation for Communication</option>
            <option value="Cyber Business Management">Cyber Business Management</option>
            <option value="Cinema and Digital Media Production">Cinema and Digital Media Production</option>
            <option value="Acting and Directing for Cinema">Acting and Directing for Cinema</option>
            <option value="Production Design">Production Design</option>
            <option value="Cinema and Digital Media Management">Cinema and Digital Media Management</option>
            <option value="Tourism Communication">Tourism Communication</option>
            <option value="Health Communication">Health Communication</option>
            <option value="Innovation Management Communication">Innovation Management Communication</option>
            <option value="Economics Communication">Economics Communication</option>
        </select>
    </div>
</div>

<div class="row">
    <div class="col-12 col-sm-6 justify-content-ceneter">
        <div class="mb-3">
            <label for="firstname" class="form-label fw-bold">First Name</label>
            <input type="firstname" class="form-control" name="name" id="firstname" aria-describedby="firstname"
                placeholder="First Name" required>
        </div>
    </div>
</div>

<div class="col-12 col-sm-6 justify-content-ceneter">
    <div class="mb-3">
        <label for="lastname" class="form-label fw-bold">Last Name</label>
        <input type="student_" class="form-control" name="lastname" id="lastname" aria-describedby="student_"
            placeholder="Last Name" required>
    </div>
</div>
</div>

<div class="row">
    <div class="col-12 col-sm-6 justify-content-ceneter">
        <div class="mb-3">
            <label for="email" class="form-label fw-bold">Email</label>
            <input type="email" class="form-control" name="email" id="email" aria-describedby="email"
                placeholder="Email" required>
        </div>
    </div>
    <div class="col-12 col-sm-6 justify-content-ceneter">
        <div class="mb-3">
            <label for="phone" class="form-label fw-bold">Phone</label>
            <input type="phone" class="form-control" name="phone" id="phone" aria-describedby="phone"
                placeholder="Phone" required>
        </div>
    </div>
</div>
</div>

<div class="row">
    <div class="col-12 col-sm-6 justify-content-ceneter">
        <div class="mb-3">
            <label for="buasri_id" class="form-label fw-bold">Buasri ID</label>
            <input type="buasri_id" class="form-control" name="buasri_id" id="buasri_id" aria-describedby="buasri_id"
                placeholder="Buasri ID" required>
        </div>
    </div>
</div>
</form>
<!-- Detail Course End -->

```

ในส่วนนี้จะเป็นหน้าของการเพิ่มนิสิตเข้าไปในระบบโดย admin ซึ่งจะใช้เพียง html อย่างเดียว

เพื่อให้กรอกข้อมูลแล้วจึง post method ไปที่ backend เพื่อเก็บข้อมูลเข้า database

```

<!-- Login Form Start -->
<section class="container-content-custom">

    <div class="d-flex h-100 justify-content-center position-relative">
        <div class="d-flex h-100 justify-content-center align-items-center">
            <div class="col-8 bg-transparent rounded">
                <div class="mb-4">
                    <h1 class="text-start text-white" style="font-weight: 500;">Student Attendance ManagementBuasri ID</label>
                        <input type="text" class="form-control shadow-sm bg-transparent" id="buasri_id" name="buasri_id" placeholder="Buasri ID">
                    </div>
                    <div class="mb-4">
                        <label for="password" class="form-label text-white">Password</label>
                        <input type="password" class="form-control shadow-sm bg-transparent" id="password" name="password" placeholder="Password">
                    </div>
                    <div class="mb-4">
                        <a href="/forget" class="text-white text-decoration-none">Forgot your password ?</a>
                    </div>
                    <button type="submit" class="btn btn-light w-100 shadow-sm" style="color: #007CB7;">Sign In</button>
                </form>
                <p class="mb-0 mt-4 text-start text-white">Not registered yet ? <a href="/signup" class="text-decoration-none text-dark">&ampnbsp Sign Up Here!</a></p>
            </div>
        </div>
    </div>
</section>
<!-- Login Form End -->

```

ในส่วนนี้จะเป็นหน้าแรกสุดของเว็บไซต์ ซึ่งคือหน้า login โดยจะใช้เพียง html เช่นเดียวกัน

ให้ผู้ใช้กรอกข้อมูลใน form แล้ว post method ไป authen ที่ฟัง backend

เพื่อตรวจสอบว่าข้อมูลนั้นถูกต้องหรือไม่

```

<!-- Table Start -->
<section>
    <div class="table-responsive" id="no-more-tables">
        <table id="myTable" class="table bg-white table-striped">
            <thead class="text-dark">
                <tr>
                    <th>Semester</th>
                    <th>Course Code</th>
                    <th>Course Name</th>
                    <th>Section</th>
                    <th>Teacher</th>
                    <th>Number of Students</th>
                    <!-- <th>Attendance Rate</th> -->
                </tr>
            </thead>
            <tbody id="tableData">
                &% for(var i=0; i < course.length; i++) { %>
                    <tr>
                        <td>&lt;= course[i].semester %></td>
                        <td>&lt;= course[i].subject_code %></td>
                        <td>&lt;= course[i].name %></td>
                        <td>&lt;= course[i].section %></td>
                        <td>&lt;= course[i].teacher %></td>
                        <td>&lt;= course[i].student_count %></td>
                    <% } %>
                </tr>
            </tbody>
        </table>
    </div>
</section>
<!-- Table End-->

```

```

<!-- Table Start -->
<section>
    <div class="mb-3 d-flex">
        <input type="text" id="myInput" name="myInput" placeholder="Search for students" class="form-control">
    </div>
    <div class="table-responsive" id="no-more-tables">
        <table id="myTable" class="table bg-white table-striped">
            <thead class="text-dark">
                <tr>
                    <th>Student ID</th>
                    <th>Student Name</th>
                    <th>Major</th>
                    <!-- <th>Attendance Count</th> -->
                    <!-- <th>Absent Count</th>
                    <th>Leave Count</th>
                </tr>
            </thead>
            <tbody id="tableData">
                &% for(var i=0; i < student.length; i++) { %>
                    <tr>
                        <td>&lt;= student[i].student_id %></td>
                        <td>&lt;= student[i].firstname %> &lt;= student[i].lastname %></td>
                        <td>&lt;= student[i].major %></td>
                        <td>&lt;a href="/students/&lt;= student[i].buasri_id %>">&lt;i class="bi bi-chevron-right fs-4" style="color: #007CB7;">&lt;/i></a></td>
                    </tr>
                &% } %>
            </tbody>
        </table>
    </div>
</section>
<!-- Table End-->

```

ในส่วนนี้จะเป็นของหน้าที่ใช้สำหรับแก้ไขรายวิชา โดยข้อมูลที่ดึงมาแสดงผลจะมี 2 ส่วน

คือรายละเอียดของรายวิชา เช่น ชื่อวิชา รหัสวิชา

และอีกส่วนหนึ่งคือข้อมูลของนักเรียนที่ลงทะเบียนเรียนในรายวิชา โดยทั้ง 2 ส่วนจะนำมาแสดงผลโดยใช้ ej

เพื่อ for loop จาก array ของข้อมูลให้ออกมาเป็น row ตามจำนวนของข้อมูลในแต่ละ table

Backend

ในส่วนของ backend จะใช้ NodeJS เป็นหลักในการทำงาน และจะใช้ library เพิ่มเติมบางส่วน

Library

```
"dependencies": {
  "bcryptjs": "^2.4.3",
  "body-parser": "1.20.1",
  "ejs": "3.1.8",
  "express": "4.18.2",
  "express-fileupload": "1.4.0",
  "express-session": "1.17.3",
  "fs": "0.0.1-security",
  "localStorage": "1.0.4",
  "mysql": "2.18.1",
  "nodemailer": "6.8.0",
  "nodemailer-smtp-transport": "2.7.4",
  "nodemon": "2.0.20",
  "path": "0.12.7"
}
```

Library ที่ใช้ทั้งหมดจะมี 13 ตัว ได้แก่

- bcryptjs : ใช้สำหรับการ hash password เพื่อเพิ่มความปลอดภัยของข้อมูลก่อนนำเข้าไปเก็บใน database
- body-parser : ใช้สำหรับการส่งข้อมูลด้วย post method
- ejs : ใช้สำหรับส่วน frontend ในการตีงข้อมูลจาก backend มาแสดงผลบนหน้าเว็บ
- express : ใช้สำหรับการเปิด server
- express-fileupload: ใช้สำหรับจัดการการ upload ไฟล์
- express-session : ใช้สำหรับการเก็บ session ของการเข้าใช้ และใช้ทำระบบ login
- fs : ใช้สำหรับการสร้างไฟล์สำหรับ download
- localStorage: ใช้สำหรับเก็บข้อมูลชั่วคราว
- mysql : ใช้สำหรับการเชื่อมต่อกับ mysql database
- nodemailer : ใช้สำหรับการส่ง email ในส่วนของ forget password
- nodemailer-smtp-transport : ใช้สำหรับการส่ง email ในส่วนของ forget password
- nodemon : ใช้สำหรับการเปิด server และการรันไฟล์
- path : ใช้สำหรับการเชื่อมและระบุ path ของไฟล์หรือ api

ตัวอย่าง API บางส่วน

```
router_db.route('/students')
.get((req,res)=>{
    connection.query('select * from users where status = "Student"',(err,result)=>{
        res.render('admin/a_all_student.ejs', {
            students: result
        })
    })
});
```

/students จะเป็นหน้าสำหรับการดูรายชื่อนิสิตทั้งหมด โดยจะเข้าได้เฉพาะ admin เท่านั้น และจะมีการ query ไปที่ database เพื่อดึงข้อมูลนิสิตออกมาแล้วไป render ejs พร้อมกับส่งข้อมูลนิสิตไป

```
router_db.route('/report')
.get((req,res)=>{
    connection.query('select status from users where buasri_id = ?', req.session.userID, (err,result)=>{
        const status = result[0].status;
        if (status == 'Student') {
            res.render('student/s_leave.ejs');
        } else if (status == 'Teacher') {
            res.redirect('/edit-report');
        }
    })
});

.post((req,res)=>{
    const subject_code = req.body.subject_code;
    const section = req.body.section;
    const type = req.body.type;

    const pdf = req.body.file;
    var blob = Buffer.from(pdf,"base64");

    connection.query('select curtime() as time, curdate() as date',(err,results)=>{
        const time = results[0].time;
        const date = results[0].date;

        const post = [
            buasri_id: req.session.userID,
            // buasri_id: req.body.buasri_id,
            subject_code: subject_code,
            section: section,
            type: type,
            file: blob,
            time: time,
            date: date,
            status: 'Pending'
        ]

        connection.query('insert into report set ?',post,(err,result)=>{
            console.log('Add report successfully');
        })
    })

    res.redirect('/courses')
});
```

/report จะใช้สำหรับแสดงผลหน้าที่นิสิตส่ง report ให้กับอาจารย์ หากอาจารย์เข้า /report จะทำการ redirect ไปที่ /edit-report แทน และสำหรับ post method จะเป็นการบันทึกข้อมูลเข้า database โดยตัวไฟล์ pdf ที่อัปโหลดเข้ามายังถูกแปลงเป็น base64 ที่ frontend และนำมาแปลงให้เป็น buffer และจึงเก็บเข้า database

```
router_db.route('/download/report-:report_id')
.get((req,res)=>{
  connection.query('select file from report where report_id = ?',req.params.report_id,(err,result)=>{
    const filebuffer = result[0].file;
    fs.writeFileSync("report.pdf", filebuffer);
    res.download('report.pdf');
  })
})
```

/download/report-:report_id จะใช้สำหรับการ download ไฟล์จาก database โดยจะดึงออกมาเป็น buffer และนำ buffer นั้นมาสร้างเป็นไฟล์โดยใช้ fs และจึงส่งให้ download ไฟล์ที่เพิ่งสร้างขึ้น

```
router_db.route('/add-courses')
.get((req,res)=>{
  connection.query('select * from users where status = "Student"',(err,result)=>{
    res.render('admin/a_add_course.ejs',{
      student: result
    })
  })
})

.post((req, res)=>{
  const semester = req.body.semester;
  const name = req.body.name;
  const subject_code = req.body.subject_code;
  const section = req.body.section;
  const teacher = req.body.teacher;

  const course = {
    semester: semester,
    name: name,
    subject_code: subject_code,
    section: section,
    teacher: teacher,
  };

  console.log(course)

  connection.query('insert into subject set ?', course, (err)=>{
    console.log('Add Course Successfully');
  });

  const students = req.body.students // array of student

  console.log(students)
  for (const stu in students) {
    const enroll = {
      subject_code: subject_code,
      section: section,
      buasri_id: students[stu],
      attendance_count: 0
    }
    connection.query('insert into enrollment set ?', enroll, (err)=>{
      console.log('Add Enrollment for',students[stu],'Successfully');
    })
  }
})
```

```

connection.query(
    'select count(buasri_id) as student_enroll from enrollment where subject_code = ? and section = ?',
    [subject_code, section],
    (err, result)=>{
        connection.query('update subject set student_count = ? where subject_code = ? and section = ?',
        [result[0].student_enroll,subject_code,section],
        (err)=>{
            |   console.log('Add No. of Student To Course Successfully');
        });
    });

    res.redirect('/courses');
);

```

/add-courses ใช้สำหรับ admin เพื่อสร้าง course หรือวิชาและเพิ่มเข้าไปใน database สำหรับ post method นั้นจะมีการเพิ่มข้อมูลเข้า database 2 ครั้ง คือการเพิ่ม course เข้าไปก่อน และจึงเพิ่ม enrollment ให้กับนิสิต โดยนำ array ของรหัสบัตรห้องนิสิตมาเพิ่มเข้าไปทีละคน จากนั้นจึง update จำนวนนักเรียนที่ส่วนของ course

```

router_db.route('/delete-students/:buasri_id')
.get((req, res)=>{
    connection.query('delete from enrollment where buasri_id = ?',req.params.buasri_id,(err)=>{
        connection.query('delete from users where buasri_id = ?',req.params.buasri_id,(err)=>{
            console.log('Delete Successfully');

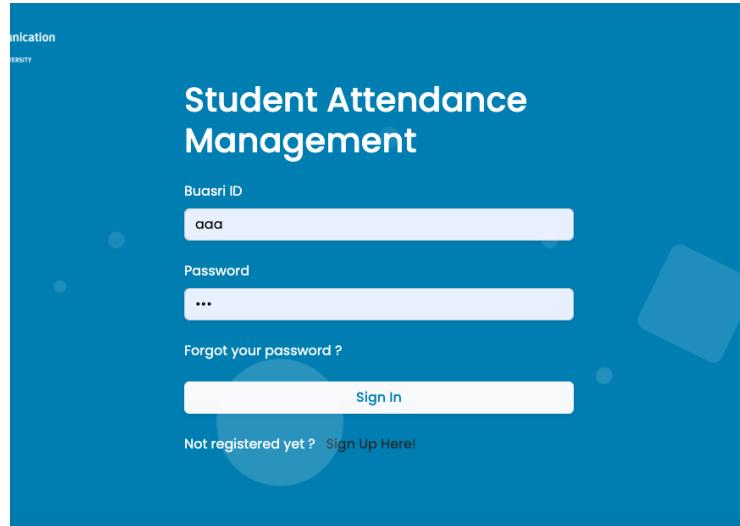
            connection.query('select subject_code, section from subject',(err, results)=>{
                for (const index in results) {
                    connection.query(
                        'select count(enroll_id) as student_enroll from enrollment where subject_code = ? and section = ?',
                        [results[index].subject_code, results[index].section],
                        (err, result)=>{
                            |   console.log(result[0].student_enroll,'enroll')
                            connection.query('update subject set student_count = ? where subject_code = ? and section = ?',
                            [result[0].student_enroll,results[index].subject_code,results[index].section],
                            (err)=>{
                                |   console.log('Update No. of Student To Course Successfully');
                            });
                        });
                }
            });
        });
    });
    res.redirect('/students');
});

```

/delete-students/:buasri_id จะใช้สำหรับ admin เพื่อลบนิสิตออกจากระบบ โดยการลบนิสิตออกได้นั้นจะต้องลบทั้งหมด 2 ครั้ง โดยจะลบจาก enrollment หรือการลงทะเบียนเรียนจากการเพิ่ม course ก่อน และจึงลบนิสิตออกจาก users นอกจากรายชื่อที่ update จำนวนนักเรียนทั้งหมดของ course ที่มีการเปลี่ยนแปลง

5. รายละเอียดโปรแกรมและวิธีการพัฒนา

Admin



```
app.get('/login', (req,res)=>{
    if (req.session.loggedin == true) {
        res.redirect('/courses')
    } else {
        // go to login page
        res.sendFile(path.join(__dirname, "/public/html/index.html"));
    }
});
```

หน้าหลักจะเป็นหน้า login ซึ่งต้องใส่ Buasri ID กับ password เพื่อ login เข้าสู่ระบบ ส่วนของ code จะเป็นการเช็คว่า ได้มีการ login หรือไม่ ถ้ามีจะ redirect เข้าเป็นหน้า /courses

```
app.post('/authen', (req, res)=>{
    var user_buasri = req.body.buasri_id;
    var user_password = req.body.password;

    if (user_buasri && user_password) {
        connection.query(
            'select * from users where buasri_id = ?', user_buasri,
            (err, results)=>{
                if (err) {console.error();}

                if (results.length > 0) {
                    bcrypt.compare(user_password, results[0].password, (err,
                    result)=>{
                        if (result == true) {
                            req.session.loggedin = true;
                            req.session.userID = user_buasri;
                            res.redirect('/courses');
                        }
                    });
                }
            }
        );
    }
});
```

โดยการ login จะส่งมาที่ authen ด้วย post method เพื่อนำข้อมูลที่กรอกไปตรวจสอบกับข้อมูลที่อยู่ใน database โดย password ที่เก็บอยู่ใน database นั้นจะถูก hash ไว้ จึงต้องใช้ bcrypt เพื่อ compare ระหว่าง password ที่ hash แล้วอยู่ใน database และ password ที่กรอกจากหน้า login ว่าตรงกันหรือไม่

Semester	Course Code	Course Name	Section	Teacher	Number of Students
2-2022	cos127	vvvv	b01	coll2	1
2-2022	cos143	asdads	b01	coll2	0
2-2022	cos145	asdads	b01	coll2	0

```
<!-- Table -->
<section>
  <div class="mb-3 d-flex justify-content-between">
    <input type="text" id="myInput" name="myInput" placeholder="Search for courses"
      class="form-control w-100 me-4">

    <a href="/add-courses"><button type="button" class="btn text-white"
      style="width: 160px; background-color: #007CB7; ">+ Add Course</button></a>
  </div>
  <div class="table-responsive" id="no-more-tables">
    <table id="myTable" class="table bg-white table-striped">
      <thead class="text-dark">
        <tr>
          <th>Semester</th>
          <th>Course Code</th>
          <th>Course Name</th>
          <th>Section</th>
          <th>Teacher</th>
          <th>Number of Students</th>
        </tr>
      </thead>
      <tbody id="tableData">
        <% for(var i=0; i < courses.length; i++) { %>
          <tr>
            <td>
              <%= courses[i].semester %>
            </td>
            <td>
              <%= courses[i].subject_code %>
            </td>
            <td>
              <%= courses[i].name %>
            </td>
            <td>
              <%= courses[i].section %>
            </td>
            <td>
              <%= courses[i].teacher %>
            </td>
            <td>
              <%= courses[i].student_count %>
            </td>
            <td>
              <a href="/courses/<%= courses[i].subject_code %>-<%= courses[i].section %>"><i class="bi bi-chevron-right fs-4"></i></a>
            </td>
          </tr>
        <% } %>
      </tbody>
    </table>
  </div>
</section>
<!-- Table End-->
```

เมื่อ login สำเร็จจะเข้ามาที่หน้า course ซึ่งจะแสดงผลวิชาที่มีทั้งหมดในระบบ (สำหรับ admin) หรือวิชาที่อาจารย์เป็นผู้สอน (สำหรับอาจารย์) หรือวิชาที่นิสิตลงทะเบียน (สำหรับนิสิต) ซึ่ง code ทางฝ่าย backend จะส่งข้อมูลวิชาไปที่หน้า frontend ซึ่งใช้ ejs ในการแสดงผลให้ออกมาเป็น table

The screenshot shows two main sections of the application:

- Teachers Section:** This section displays a table of teachers with columns for Name, Email, and Phone. A single row is shown with the name "ooo", email "sa@g.com", and phone "1234567891". A blue button labeled "+ Add teacher" is visible.
- Students Section:** This section displays a table of students with columns for Student ID, Name, Major, Email, and Phone. Three rows are shown with student IDs 12345, 12345, and 13123, names aaaaa, aaaaa, and asdasd, majors commu, commu, and commu, emails asd@gmail.com, asd@gmail.com, and koppok32a@gmail.com, and phones 01234, 01234, and 123123 respectively. A blue button labeled "+ Add student" is visible.

โดยสำหรับ admin จะสามารถดูข้อมูลของอาจารย์ทั้งหมดและนิสิตทั้งหมดได้ ซึ่งใช้วิธีเดียวกันกับการแสดงผลวิชาทั้งหมด ซึ่งคือการใช้ ejs นำข้อมูลที่ได้จาก database มาแสดงผลเป็น table

The screenshot shows two sections of the application:

- Courses > Add Course Section:** This section contains forms for adding a new course. It includes fields for Semester (Semester), Course Name (Course Name), Teacher ID (Teacher ID), Course Code (Course Code), Course Section (Course Section), and a "Submit" button.
- Add Students Section:** This section contains a table for adding students to a course. It includes a search bar for students, a table with columns for Buasri ID, Student Name, and Major, and a checkbox for selecting students. One row is shown with Buasri ID 222, Student Name sirisorn, and Major comm.

```

<!-- Course Details Start -->


<form method="POST" action="/add-courses">
    <div class="my-4 d-flex justify-content-between align-items-center">
      <h1>Courses > Add Course</h1>
      <button type="submit" class="btn text-white h-50"
        | style="width: 100px; background-color: #007CB7;">Submit</button>
    </div>

    <!-- Text Field Start -->
    <div class="row">
      <div class="col-12 col-sm-6 justify-content-md-ceneter">
        <div class="mb-3">
          <label for="semester" class="form-label fw-bold">Semester</label>
          <input type="text" class="form-control" name="semester" id="semester" aria-describedby="semester"
            placeholder="Semester">
        </div>
      </div>

      <div class="col-12 col-sm-6 justify-content-md-ceneter">
        <div class="mb-3">
          <label for="subject_code" class="form-label fw-bold">Course Code</label>
          <input type="text" class="form-control" name="subject_code" id="subject_code" aria-describedby="subject_code"
            placeholder="Course Code">
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-12 col-sm-6 justify-content-md-ceneter">
        <div class="mb-3">
          <label for="name" class="form-label fw-bold">Course Name</label>
          <input type="text" class="form-control" name="name" id="name" aria-describedby="name"
            placeholder="Course Name">
        </div>
      </div>

      <div class="col-12 col-sm-6 justify-content-md-ceneter">
        <div class="mb-3">
          <label for="section" class="form-label fw-bold">Course Section</label>
          <input type="text" class="form-control" name="section" id="section" aria-describedby="section"
            placeholder="Course Section">
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-12 col-sm-6 justify-content-md-ceneter">
        <div class="mb-3">
          <label for="teacher" class="form-label fw-bold">Teacher ID</label>
          <input type="text" class="form-control" name="teacher" id="teacher" aria-describedby="teacher"
            placeholder="Teacher ID">
        </div>
      </div>
    </div>
  </form>
  <!-- Text Field End -->


```



```

<!-- Add Student Start -->
<div class="mt-5 mb-3 d-flex justify-content-between align-items-center">
  <h3>Add Students</h3>
</div>

<!-- Table -->
<section>
  <div class="mb-3 d-flex">
    <input type="text" id="myInput" name="myInput" placeholder="Search for students"
      | class="form-control">
  </div>
  <div class="table-responsive" id="no-more-tables">
    <table id="myTable" class="table bg-white table-striped">
      <thead class="text-dark">
        <tr>
          <th>Buasri ID</th>
          <th>Student Name</th>
          <th>Major</th>
        </tr>
      </thead>

```

```

<tbody id="tableData">
    <% for(var i=0; i < student.length; i++) { %>
    <tr>
        <td>
            <%= student[i].buasri_id %>
        </td>
        <td>
            <%= student[i].name %>
        </td>
        <td>
            <%= student[i].major %>
        </td>
        <td>
            <div class="form-check">
                <input class="form-check-input" name="students[]" type="checkbox" value="<%= student[i].buasri_id %>" id="defaultCheck1">
            </div>
        </td>
    </tr>
    <% } %>
</tbody>
</table>
</div>
</section>
<!-- Table End-->

```

Students > Add student

Student ID **Major**

First Name **Last Name**

Email **Phone**

Buasri ID

Submit

Teachers > Add teacher

First Name **Last Name**

Email **Phone**

Buasri ID

Submit

Teachers > Name > Edit

First Name: ooo

Last Name: AAA

Email: sa@g.com

Phone: 1234567891

Delete submit

นอกจากนี้ admin ยังสามารถ CRUD หรือว่า create, read, update, delete ข้อมูลทั้งหมดใน database ได้ เพื่อจัดการ เพิ่ม และแก้ไขรายชื่อวิชา นิสิต และอาจารย์

อาจารย์

Courses

Semester	Course Code	Course Name	Section	Number of Students		
2-2022	cosl27	vvvv	b01	1	View	Edit
2-2022	cosl43	asdads	b01	0	View	Edit
2-2022	cosl45	asdads	b01	0	View	Edit

ในหน้าหลังจาก login เข้ามาของอาจารย์จะเป็นหน้า course เช่นเดียวกัน
แต่จะแสดงผลเฉพาะวิชาที่อาจารย์สอนเท่านั้น ไม่แสดงผลวิชาอื่นที่อาจารย์ไม่ได้สอน
และจะมีปุ่มที่สามารถกดเพื่อเริ่มการเช็คชื่อเข้าเรียนเพื่อไปยังหน้าเช็คชื่อได้

 College of
Social Communication
Innovation
SRINAKHARINWIROT UNIVERSITY

Courses Leave Requests | 

Courses > Details



Semester	Course Code	Course Name	Section	Teacher	Number of Students
2-2022	cosi27	vvvv	b01	coil2	1

Students

Search for students

Student ID	Student Name	Major	Attendance Count
12345	aaaaa wit	commu	0

อาจารย์สามารถกดปุ่มทางขวาสุดของแต่ละวิชาเพื่อเข้าไปดูข้อมูลรายละเอียดของวิชาแต่ละวิชาได้ และสามารถกดได้ว่ามีนิสิตคนใดลงทะเบียนเรียนวิชานั้นบ้าง และสามารถกดจำนวนการเข้าเรียนของนิสิตแต่ละคนได้

 College of
Social Communication
Innovation
SRINAKHARINWIROT UNIVERSITY

Courses Leave Requests | 

Courses > Start Attendance



22:39:25

ในส่วนนี้จะเป็นการสร้าง QR Code สำหรับใช้เพื่อเข้าเรียน โดยระบบจะทำการ generate QR Code ใหม่ทุกรั้งตามเวลาที่กำหนด หากเลยเวลาที่กำหนด QR Code จะไม่สามารถใช้งานได้ ภายใน QR Code ได้เก็บข้อมูลรหัสวิชา, Section, วันและเวลาที่สร้าง QR Code, รหัส QR Code

ซึ่งข้อมูลเหล่านี้ถูกเก็บในรูปแบบของ Path สำหรับส่งข้อมูลเป็น Parameter ไปยัง Backend จากนั้นจึงนำข้อมูลเหล่านี้เก็บลง Database

ฟังก์ชันสำหรับเรียกดูข้อมูลวันที่

```
//set up date
function formatDate() {
    var d = new Date(),
        month = '' + (d.getMonth() + 1),
        day = '' + d.getDate(),
        year = d.getFullYear();

    if (month.length < 2)
        month = '0' + month;
    if (day.length < 2)
        day = '0' + day;

    // return [year, month, day].join('-');
    return `${year}:${month}:${day}`
}
```

ฟังก์ชันสำหรับเรียกดูข้อมูลเวลา

```
//set up time
function formatTime() {
    var dateTime = new Date();
    var hrs = dateTime.getHours();
    var min = dateTime.getMinutes();
    var sec = dateTime.getSeconds();

    if (hrs < 10) {
        hrs = '0' + hrs;
    }

    if (min < 10) {
        min = '0' + min;
    }

    if (sec < 10) {
        sec = '0' + sec;
    }

    var time = `${hrs}:${min}:${sec}`

    return time;
}
```

ฟังก์ชันแสดงเวลาบนหน้าเว็บ โดยใช้ setInterval() ในการอัพเดตข้อมูลทุก 1 วินาที

```
//display time on screen
function displayTime() {
    const time_label = document.getElementById("time_label");
    time_label.innerHTML = formatTime();
}
setInterval(displayTime, 1000);
```

ฟังก์ชันสำหรับใช้ในการ Generate ID ให้กับ QR Code

```
//generate for making unique path
function generateID() {
    var string = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    let id = '';

    // Find the length of string
    var len = string.length;
    for (let i = 0; i < 6; i++) {
        id += string[Math.floor(Math.random() * len)];
    }
    console.log("id generation:", id);
    return id;
}
```

ฟังก์ชันสำหรับ Generate Path ซึ่งเป็นการนำเอา ID ที่ได้มาอัพเดตแล้วส่งเป็น Parameter ไปยัง Backend

```
//create path that links to student attendance page
function generatePath(path_id) {
    let dateTime = {
        date: formatDate(),
        time: formatTime()
    }

    let subject_code = "COS101";
    let section = "B02";

    // let api = 'http://localhost:5005/attendance-check/:subject_code-:section/:date-:time/:path_id'
    let path = `http://localhost:5005/attendance-check/${subject_code}-${section}/${dateTime.date}-${dateTime.time}/${path_id}`

    return path
}
```

ฟังก์ชันสำหรับการนำเอา API มาแปลงเป็น QR Code โดยจะทำการแปลงให้เป็น base64 ก่อนจากนั้นใช้ Google Chart API ในการสร้างให้เป็น QR Code ส่งกลับมา

```
//encode qrcode into base64
function htmlEncode(value) {
    return $('<div/>').text(value)
        .html();
}

//set base64 as img
function updateQRCode() {
    let path_id = generateID();
    let finalURL =
        'https://chart.googleapis.com/chart?cht=qr&chl=' +
        htmlEncode(generatePath(path_id)) +
        '&chs=160x160&chld=L|0'

    $('.qr-code').attr('src', finalURL)
    $.post("/attendance-create/test",
    {
        path_id: path_id
    },
    function (data, status) {
        console.log(data);
    });
}

// attendance_rate = httpGet(`get-live-rate/${class_id}`)
}

setInterval(updateQRCode, 30000);
```

The screenshot shows a web-based application interface. At the top, there's a blue header bar with the college logo on the left and navigation links for 'Courses', 'Leave Requests', and a user profile icon on the right. The main content area has a light gray background. A large heading 'Leave Requests' is centered at the top of the content area. Below it is a search bar with the placeholder text 'Search for courses'. Underneath the search bar is a table with the following columns: Student ID, Student Name, Course Name, Section, Leave Type, Leave Date, File, and Status. The table currently has no data rows.

ในส่วนของหน้า leave request จะเป็นการดูข้อมูลใบลาต่าง ๆ ที่นิสิตส่งมาได้ โดยจะมีสถานะขึ้นบวกกว่าเป็น pending, approve, reject และสามารถแก้ไขสถานะของใบลาได้

นิสิต

The screenshot shows the student dashboard interface. At the top, there is a logo for the College of Social Communication Innovation, Srinakharinwirot University. To the right of the logo are navigation links for "Courses" and "Leave Application". There is also a user profile icon. Below the header, the word "Courses" is prominently displayed. A search bar labeled "Search for courses" is present. To the right of the search bar is a blue button labeled "Take Attendance" with a QR code icon. A table below the search bar lists course details:

Semester	Course Code	Course Name	Section	Teacher
2-2022	cos127	vvvv	b01	coll2

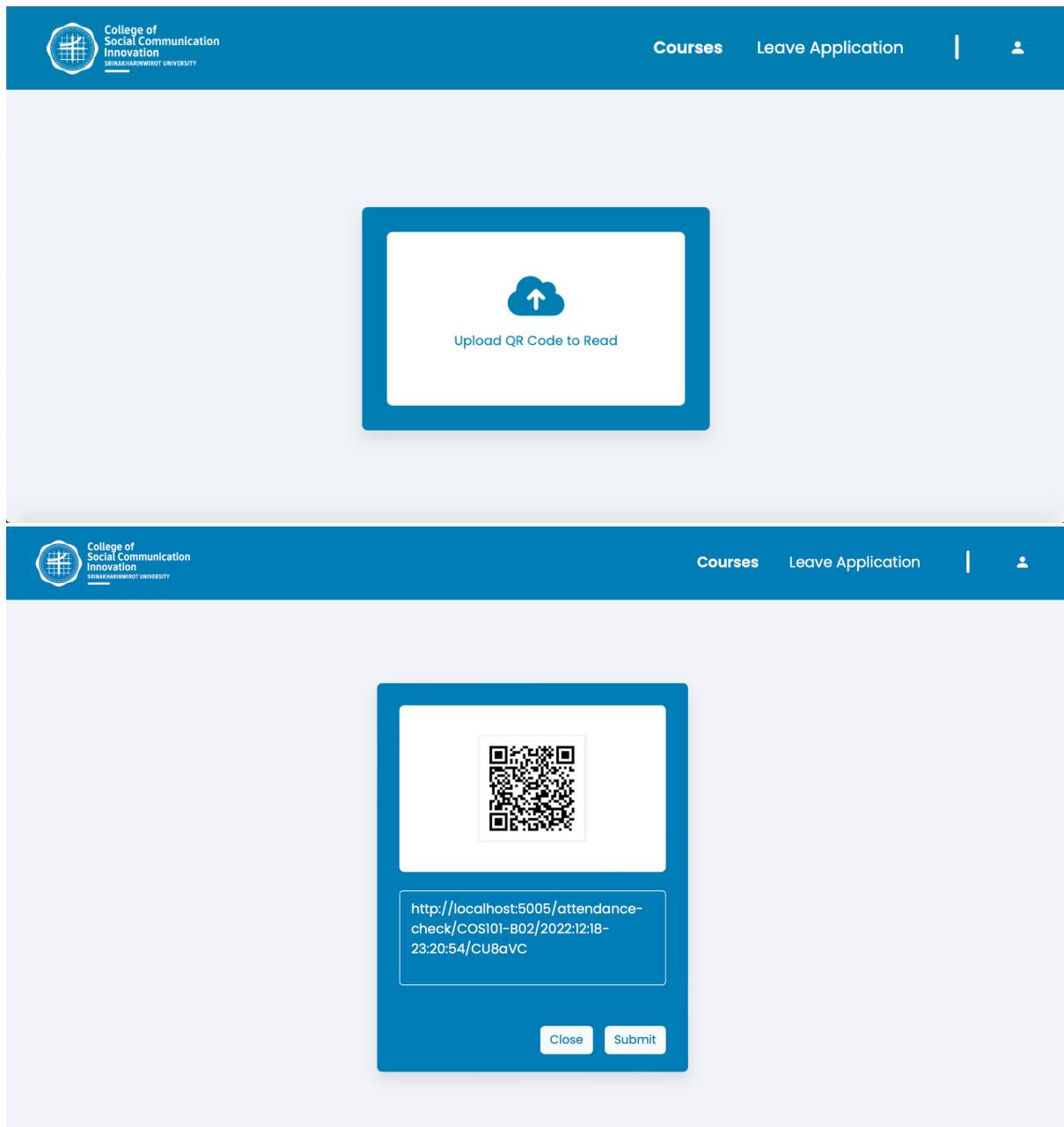
ในส่วนของนิสิต หน้าที่เข้ามาหลังจาก login จะเป็นหน้า course เช่นเดียวกัน แต่จะแสดงผลเฉพาะวิชาที่นิสิตลงทะเบียนเรียนเท่านั้น โดยจะมีปุ่มสำหรับกรอกข้อมูลเช็คชื่อเข้าเรียน

The screenshot shows the student dashboard interface. At the top, there is a logo for the College of Social Communication Innovation, Srinakharinwirot University. To the right of the logo are navigation links for "Courses" and "Leave Application". There is also a user profile icon. Below the header, the word "Courses > Details" is displayed. A search bar labeled "Search for students" is present. Below the search bar, there is a table with columns: Semester, Course Code, Course Name, Section, Teacher, and Attendance Count. The data shown is:

Semester	Course Code	Course Name	Section	Teacher	Attendance Count
1-2022	cos123	ooo	B01	2345	0

Below the table, the section "Attendance" is shown. It includes a search bar for students and three columns: "Attendance Date", "Attendance Time", and "Attendance Image".

นิสิตสามารถเข้ามาดูรายละเอียดของแต่ละวิชาได้ เช่นเดียวกัน โดยจะสามารถดูได้ว่าเข้าเรียนทั้งหมดกี่ครั้ง เข้าเรียนแต่ละครั้งที่เวลาใด และสามารถ download รูปที่ถ่ายเป็นหลักฐานการเข้าเรียนได้



หน้า Scan QR Code เนื่องจากว่าใช้คอมพิวเตอร์ในการทดสอบ ดังนั้นจึงไม่สามารถใช้กล้องในการสแกนได้
จึงทำการสร้างระบบ Scan QR Code ขึ้นมาเพื่อรับการสแกน QR Code ซึ่งเมื่อ Scan
ออกมายจะเห็นได้ว่าจะได้ข้อมูลเป็น API ออกมา และเมื่อกด Submit
แล้วก็จะไปยังหน้ากรอกข้อมูลยืนยันการเข้าเรียนต่อไป

พังก์ชั่นในการเรียกใช้ API จากภายนอก สำหรับการอ่าน QR Code

```

const wrapper = document.querySelector(".wrapper"),
  form = document.querySelector("form"),
  fileInp = form.querySelector("input"),
  infoText = form.querySelector("p"),
  closeBtn = document.querySelector(".close"),
  submitBtn = document.querySelector(".submit");
  modalTitle = document.querySelector(".modal-title");
  modalBody = document.querySelector(".modal-body")

function fetchRequest(file, formData) {
  infoText.innerText = "Scanning QR Code...";

  fetch("http://api.qrserver.com/v1/read-qr-code/", {
    method: 'POST', body: formData
  }).then(res => res.json()).then(result => {
    result = result[0].symbol[0].data;
    infoText.innerText = result ? "Upload QR Code to Scan" : "Couldn't scan QR Code";
    if (!result) return;
    document.querySelector("textarea").innerText = result;
    form.querySelector("img").src = URL.createObjectURL(file);
    wrapper.classList.add("active");
    submitBtn.href = result;
  }).catch(() => {
    infoText.innerText = "Couldn't scan QR Code";
  });
}

```

Attendance

Course Code: swu11

Course Section: B01

Attendance Date: 18/12/2022

Attendance Time: 23:11:20

Take a photo

Turn on webcam Take Screenshot

ในหน้านี้จะใช้สำหรับการเช็คชื่อ โดยนิสิตจะต้องกรอกข้อมูลทุกอย่างให้ครบถ้วน และถ่ายรูปยืนยันเป็นหลักฐานการเข้าเรียน และนอกจากนี้ทางหน้าเว็บไซต์ก็จะมีการเช็คตำแหน่งของนิสิตว่าอยู่ในบริเวณของตึกเรียนหรือไม่ เพื่อเป็นการยืนยันว่าได้เข้าเรียนจริง พงกชั้นสำหรับใช้ในการตรวจสอบตำแหน่งสถานที่ ณ ปัจจุบันของผู้ใช้งาน

```

const cosci_location = {
  latitude: 13.747108408571727,
  longitude: 100.56546691267371
}

const checklocation = (student_latitude, student_longitude) => {
  console.log(student_latitude,student_longitude)
  if (cosci_location.latitude == student_latitude && cosci_location.longitude == student_longitude) {
    console.log('complete');
  }
  else if (student_latitude == 13.7366 && student_longitude == 100.4995) {
    console.log('yes this is my house, this is for testing');
  }
  else {
    console.log('fail');
    modalTitle.innerText = "Don't cheat"
    modalBody.innerText = "You are not allowed to check attendance";
  }
}

const getlocation = () => {
  fetch("https://ipapi.co/json/")
    .then((response) => response.json())
    .then((data) => {
      checklocation(data.latitude, data.longitude);
    })
}
// console.log(student_latitude,student_longitude)

```

The screenshot shows a web application interface for leave applications. At the top, there is a dark header bar with the text 'College of Social Communication Innovation' and 'SOMMERWICH UNIVERSITY'. To the right of the header are links for 'Courses', 'Leave Application', and a user profile icon. Below the header is a light blue navigation bar with the title 'Leave Application' on the left and a 'Submit' button on the right.

The main form area contains several input fields:

- Course ID:** A text input field.
- Course Section:** A dropdown menu labeled 'Open this select section'.
- Leave Type:** A dropdown menu labeled 'Open this select leave type'.
- Leave Date:** A date input field labeled 'dd/mm/yyyy'.
- Upload Leave Letter:** A file upload input field with a 'Choose File' button and a placeholder 'No file chosen'.

นิสิตสามารถส่งใบลาป่วยหรือลาภัยให้กับอาจารย์ได้ โดยจะต้องเลือกวิชาให้ถูกต้อง และเลือก upload pdf file ที่เป็นใบลาด้วย