Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 1194-1209
© UniversitiTun Hussein Onn Malaysia Publisher's Office

**AITCS**

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

# Student Attendance System using Facial Recognition

**Muhammad Ikhwan Zainal Ariffin**[1], **Mohamad Firdaus Ab. Aziz** [1]*

[1]Faculty of Computer Science & Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.069
Received 09 August 2022; Accepted 11 June 2023; Available online 30 June 2023

**Abstract**: Student Attendance System using Facial Recognition is a web application that helps both lecturer and student to record attendance while minimizing time consumption during the recording process. Traditionally, recording attendance consumes a significant amount of time because it involves each student to sign their attendance on a piece of paper. Thus, the utilization of facial recognition technology can help in reducing the time used to record attendance during teaching and learning process. Additionally, the proposed system can reduce the chances of student not getting attendance recorded when they attend a class. The methodology applied for the system is Spiral Model. Software such as Visual Studio Code, PostgreSQL, Django are used to develop the system. The system is expected to minimize the time used and error when recording attendance.

**Keyword**: Facial Recognition, Artificial Intelligence, Web Application

## 1. Introduction

Attendance is an essential element of a daily evaluation process in a classroom. This is due to the fact that whenever a student comes to a class, they must have a proof to indicate that they have attended the class. Usually, the teacher or lecturer will mark the attendance for each student but sometimes the students have to mark their own attendance by signing on a piece of paper. An attendance system using facial recognition could be used as another medium to take students' attendance. This is because it is faster and more efficient than the traditional or current method. The current method either requires each student to sign their attendance on a paper that has to be passed from a student another or it requires the teacher to mark each student attendance. This method of marking attendance is less efficient and slower compared to attendance system using facial recognition.

The current method either requires each student to sign their attendance on a paper that has to be passed from a student another or it requires the teacher to mark each student attendance. This method

*Corresponding author: mdfirdaus@uthm.edu.my* of marking attendance is less efficient and slower compared to attendance system using facial recognition.

Although a teacher or lecturer may independently confirm their students' attendance, it would be extremely difficult to verify all students in a large class. For instance, if the total number of students in the class is over 80, it would be difficult for a lecturer to determine the number of students who are absent since they may not recognize all of their students. Consequently, errors or mistakes may occur, such as when a student is actually absent, but the lecturer incorrectly marks the absent student as if the student has attended the class.

Another issue caused by the traditional attendance system is 'ghost attendance'. Ghost attendance is when a student appears as attended in the attendance record but in reality, someone such as their friend was the one who marked or signed their attendance for them. As a result, the absent student can easily escape from their punishment.

Time is an important part of our life and by using the traditional method, that is known as attendance system based on signature, more time has to be consumed just to verify the attendance of the students. This may cause inefficiency during teaching and learning session because less time could be used for the class. As an example, if a class only has an hour of session but takes 15 minutes to record the attendance, only 45 minutes can be spent during the class. The lecturer might not have enough time to teach, and the students might also not have enough time to ask the lecturer about what they do not understand. So, the topic that was taught during the class has to be carried to the next class.

Therefore, the proposed system can help eliminate the issues since it reduces the overall time taken to record attendance. A large class is also not going to take too much time as compared to the traditional method because the system can detect and record multiple students at once, thus making the recording process quicker. Next, since the system requires the student's face to record their attendance, cheating the attendance is not going to be easy so the system also reduces 'ghost attendance' issues.

This article contains five sections. Section 1 gives project overview. Section 2 explains the related works. Section 3 describes the methodology while Section 4 summarizes the result. Section 5 conclude the discussion.

## 2. Related Works

### 2.1. Machine Learning

Machine learning is an expanding branch of computational algorithms that are supposed to mimic human intelligence by learning from their surroundings [1]. Machine learning (ML) is sometimes referred to as the scientific study of algorithms and statistical models that computer systems use to carry out a job without being explicitly programmed [2]. There many algorithms related to machine learning and **Figure 1** shows some of the commonly used ones.

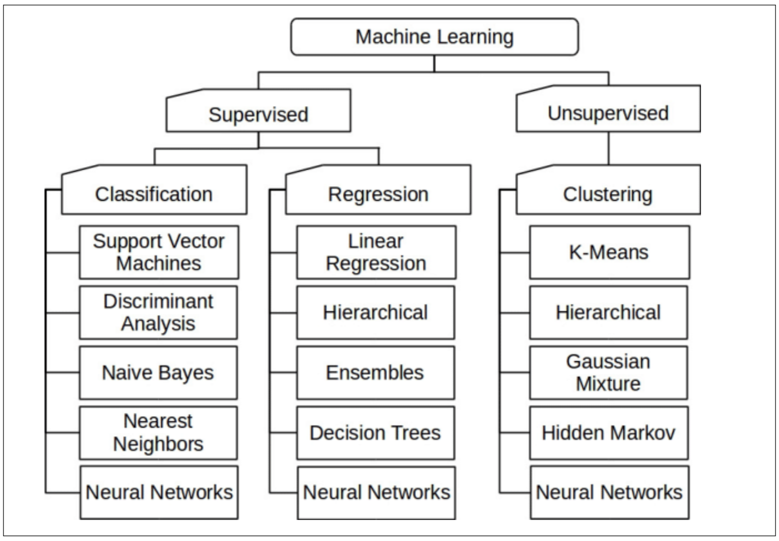


**Figure 1: Common Machine Learning algorithms**

In the 1950s, Arthur Samuel of IBM developed a computer program for playing checkers. Samuel initiated alpha-beta pruning since the program had a limited quantity of accessible computer memory. His design has a scoring system depending on the placement of the pieces on the board. The scoring function makes an effort to determine each team's probability of victory. The program chooses its next step using a minimax method, which evolved into the minimax algorithm. Samuel also designed a range of ways to aid in the improvement of his program. In a process Samuel referred to as rote learning, his program stored/remembered all prior locations and blended them with the values of the reward function. Arthur Samuel coined the phrase "machine learning" in 1952 [3].

Machine learning algorithms can carry out a task without being expressly trained to do so. It is a process in which computers are trained to perform specific tasks using data that they have been given. Algorithms can be written that inform computers how to solve simple problems; the computer doesn't have to learn anything in order for this process to work. Manually developing algorithms for increasingly complex tasks might be difficult. Rather than relying on human programmers to specify every step, it may prove more efficient to let the machine figure out its own algorithm [4].

Machine learning plays its role in Student Attendance System using Facial Recognition by having the system to learn each student’s face. The system then stores the information in the database and when the camera detects the student’s face, face comparison occurs. This is crucial because the system only record the student’s attendance after detecting their face.

## 2.2. Facial Recognition

In computing, facial recognition system is a device which matches a person’s face captured in a digital photograph or video frame against a database of human faces to identify the person. Because it works by finding and quantifying face traits from a given picture, this technology is often used to authenticate individuals via ID verification services. Facial recognition program was first created in the 1960s. Woody Bledsoe, Helen Chan Wolf, and Charles Bisson worked together to construct a computer capable of recognizing human faces. Because the coordinates of an image's facial traits had to be set by a human before the computer could use them for identification, their early facial recognition project was dubbed "man-machine." A person had to locate the coordinates of face characteristics such pupil centers, inner and outside corner of eyes, and the widows peak in the hairline on a graphics tablet. The coordinates were utilized to compute 20 different distances, including the mouth and eye widths. In this way, a person might analyze around 40 photos every hour and construct a database of determined distances. The distances for each image would then be automatically compared, the difference calculated, and the closed records returned as a probable match [5].

Facial recognition contributes its task in the system by detecting and comparing the student's face captured from the camera and the information of the faces in the database of the system. After detecting comparing the faces, the system will make the decision to either mark the student as present or absent depending on whether the system detects the student's face or not.

### 2.3. System Comparison

Face Attendance is a time clock for workers that uses facial recognition technology to check and record their timesheets. Employers and supervisors can quickly turn any Android-powered smartphone into a clocking station using Face Attendance; recorded data is then moved to the cloud, where it can subsequently be seen on any internet browser [6]. The mobile application is fully cloud-based, which means it removes the need to maintain either hardware or software. Next, it can even function when internet connection is unavailable. The data will be synchronized once the device is connected with internet connection.

AttendLab is another mobile application to record employee attendance. Admin can easily track the employee's activities such as when they clock in or out. The application supports multiple time zones attendance, so it helps companies that have global branches and it also performs the time zones conversion.

XmartClock records workers attendance by allowing them to punch-in from their mobile phone or tablet. Attendance report can be generated and employers can easily be informed regarding their workers. With this application, custom notes can be viewed with each punch.

Therefore, the comparison tells us that the proposed system has an extra feature that other compared systems do not have. The proposed system has both automatic attendance record module, which is the facial recognition attendance record module, and manual attendance record module. The manual attendance record module helps in attendance recording process if the facial recognition module is not working as it should be.

**Table 1: Comparison between existing systems and proposed system**

| Module | System | | | |
|---|---|---|---|---|
| | Face Attendance | AttendLab | XmartClock | Student Attendance System using Facial Recognition |
| System type | Android | Android | Android and iOS | Web-based |
| 1. Login module | Available | Available | Available | Available |
| 2. Registration module | Available | Available | Available | Available |
| 3. Facial recognition attendance record module | Available | Available | Available | Available |
| 4. Manual attendance record module | Unavailable | Unavailable | Unavailable | Available |
| 5. Software development | - | - | - | Spiral model |
| 6. User management module | Available | Available | Available | Available |

| | | | | | |
|---|---|---|---|---|---|
| 7. | Server-side Programming language | Python | Python | Python | Python |

## 3. Methodology

Spiral model is suitable for the project and it has the potential finish the system efficiently without using too much time to develop it. By using Spiral model, developers could communicate with users, particularly the lecturers in order to collect as much important information needed so that they will meet the requirements.

To develop Attendance System using Facial Recognition, Spiral Model has been decided to be used. The spiral model is an excellent technique that utilizes parts of both design and prototype in phases to integrate the benefits of both top-down and bottom-up approaches. It's a meta-model, which implies other models may utilize it. It also emphasizes on risk assessment and project risk reduction. This may be accomplished by breaking down a large project into smaller pieces, which may ease the adjustment process in the development phase while also allowing for risk evaluation and project continuance considerations throughout the life cycle. In this paradigm, the development team begins with a limited list of requirements and then goes through each development step for those requirements. Subsequently, the development team can learn new lessons from the first version. Additionally, the team will add functionality for new needs in ever-increasing "spirals" until the program is ready for installation and maintenance. Each quadrant prior to the production version is known as a prototype in this paradigm [7].

**Table 2: Project Description**

| Phase | Description | Findings |
|---|---|---|
| Planning | • Obtain students' information needed to develop the program | • Gantt chart<br>• Student's information |
| Risk Analysis | • Reduce risk in system | • List of possible issues |
| Development Phase | • Include the development of the proposed system<br>• Programs related with machine learning algorithms | • Interface of the system<br>• Database design |
| Evaluation | • Evaluate the current system<br>• Look for issues | • List of things to update and fix |

### 3.1. Planning Phase

The Spiral approach requires constant communication between the user, in this case the lecturer, and the system analysts to develop the system needs. Students' information, such as name, matric number, photo of faces, and more, must be gathered before the Attendance System using Facial Recognition can be developed. **Table 3** shows the system modules.

**Table 3: System modules**

| No. | Modules | Functionalities | Users |
|---|---|---|---|
| 1. | Registration Module | Allows lecturers and students to register | Lecturer, Student |
| 2. | Login Module | Allows lecturers and students to log into the system | Lecturer, Student |
| 3. | Attendance Log Module | Allows the lecturer to check attendance log | Lecturer |

| | | | |
|---|---|---|---|
| 4. | Facial Recognition Module | Detects faces in the database and records the attendance | Lecturer |
| 5. | Subject registration module | Allows students to register their subjects | Student |
| 6. | Student update module | Allows students to update their information and subjects they take | Student |

The functional requirement defines the functionalities for the system in which it converts the input obtained into output. **Table 4** shows the suggested functional requirements for the system.

**Table 4: Functional requirements**

| No. | Modules | Explanation |
|---|---|---|
| 1. | Registration Module | • System should allow user to register into the system<br>• System only allows user to register with the right credentials<br>• System will show errors if user put in the wrong input |
| 2. | Login Module | • System only takes username and password to login<br>• System displays error if the wrong input in inserted<br>• System will redirect to the main page when successful |
| 3. | Attendance Log Module | • System shows lecturer the name of student who attend or absent for the class chosen<br>• Lecturer can search the name or matric number of any student to see directly whether they attended the class or not |
| 4. | Facial Recognition Module | • System will detect faces in database and record it in the attendance database |
| 5. | Subject registration module | • System allows student to register the subject they take<br>• System allows student to register the section for their subject |
| 6. | Student update module | • System allows student to update their name<br>• System allows student to update their matric number<br>• System allows students to update their subjects<br>• System allows students to update their e-mail address |

Non-functional requirements describe how the system should work where it is not connected directly to functionality but rather to the system's usability. These requirements pertain towards the overall quality and attributes of the system. **Table 5** shows the non-functional requirement for the system.

**Table 5: Non-functional requirements**

| No. | Requirement | Explanation |
|---|---|---|
| 1. | Usability | The overall process of the system is simple to grasp. To understand how to operate the system, the user does not need to thoroughly study its general functioning. |
| 2. | Security | Correct login credentials are required by the system in order to grant user access to data connected to the user account. User that inserts wrong login credentials |
| 3. | Performance | When running the system, a reasonable reaction time is to be anticipated. |
| 4. | Availability | The system may be accessed at any time. |
| 5. | Maintainability | Maintenance will be performed at regular intervals to guarantee that all system features operate with minimal faults. |
| 6. | Integrity | The system will save information properly, ensuring that it is neither corrupted nor unreadable. |

User requirement defines what the users of the system expect from the system. **Table 6** shows the user requirement for the proposed system.

**Table 1: User requirement for the proposed system**

| No. | User requirements |
|---|---|
| i. | Lecturer can input their username and password to login |
| ii. | Student should be able to register themselves into the system |
| iii. | Student should be able to choose their subjects |
| iv. | Student should be able to upload their profile picture |
| v. | Lecturer should be able to start the facial recognition attendance record process |
| vi. | Lecturer should be able to collect the recorded information of students |
| vii. | Lecturer should be able to choose which class they want to capture |
| viii. | Lecturer should be able to generate warning letter |
| ix. | Lecturer should be able to update their information |
| x. | Lecturer should be able to delete or add attendance |
| xi. | Lecturer should be able to view the account of any student of their choice |

## 3.2. Risk Analysis Phase

Phase two of the spiral model involves risk analysis. This phase is done to identify risk and other solutions that aid in the creation of the prototype. So, other attendance choices will take more time for lecturers to record student attendance, so they are removed to decrease risk. Students may also try to circumvent the present attendance system by having a buddy report their attendance. This occurs regularly, but catching the cheater and other suspicions is difficult.

### System Analysis

The system incorporates object-oriented approach where the diagrams for UML are produced as follows.

### Use Case Diagram

A use case diagram is a graphical representation of how system elements interact. The objective of a use case diagram is to explain how the system's components relate to one another as well as the intended behavior of the system. It also demonstrates other methods in which the user may engage with the system. The use case diagram for the Student Attendance System with Facial Recognition is shown in **Figure 2**.

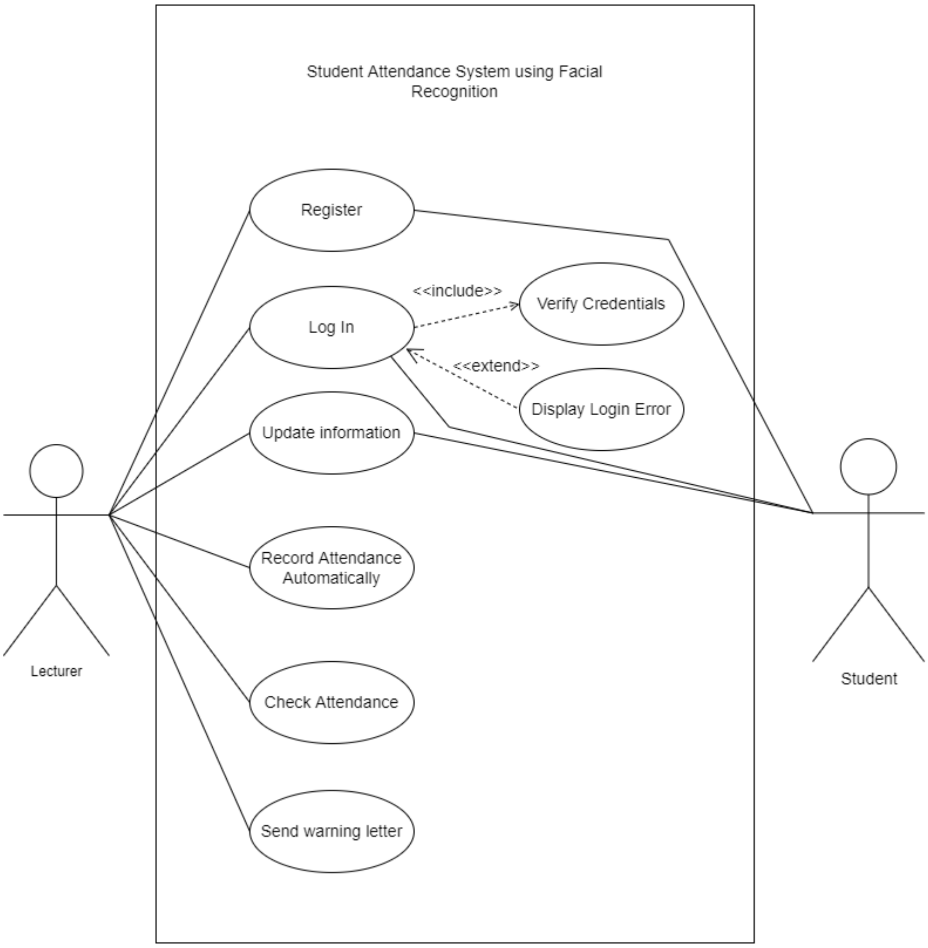


**Figure 2: Use Case Diagram**

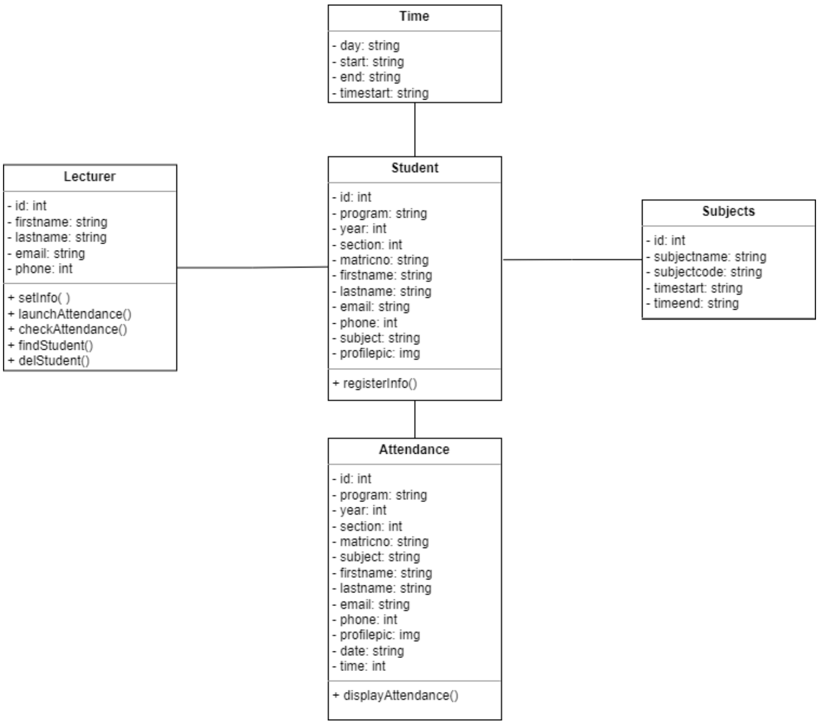


**Figure 3: Class Diagram**

**Figure 3** shows the class diagram for Student Attendance System using Facial Recognition. It includes several main classes namely, Lecturer, Student and Attendance

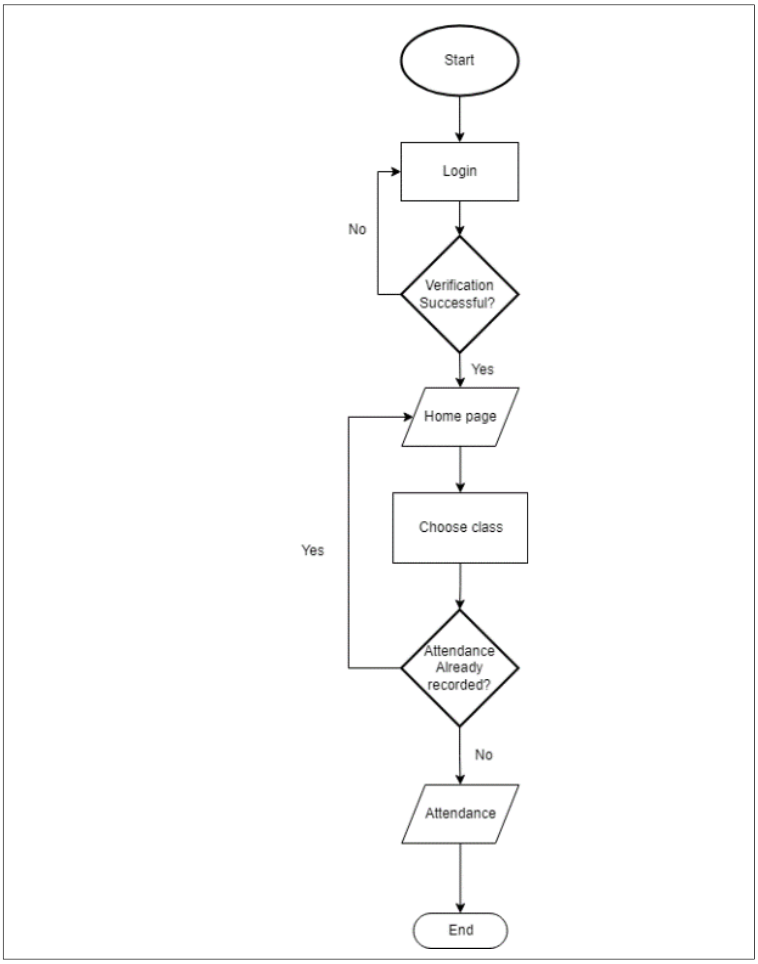


**Figure 4: System Flowchart**

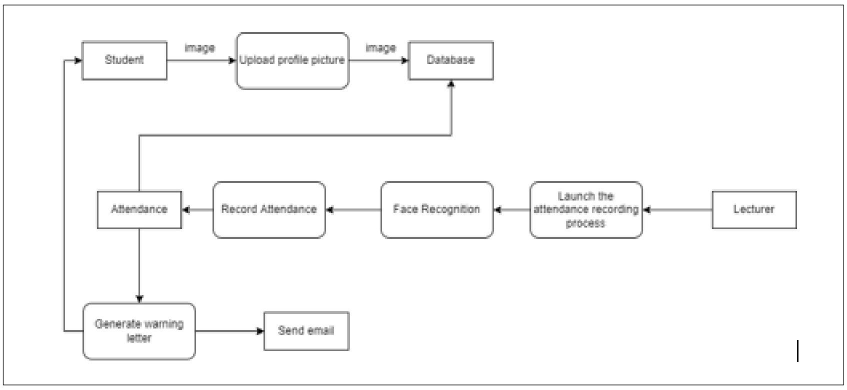


**Figure 5: System Flowchart**

### 3.3. Development Phase

The web application is produced and tested in the third quadrant of the Spiral model. Attendance System using Facial Recognition will use Python as one of its web application languages because Python has proven to be reliable and popular for projects involving artificial intelligence, especially machine learning and facial recognition. The web application's design is also improved from the prototype. This step includes testing to identify if any errors need to be fixed. For example, registration, login, and some facial recognition features are developed and tested at the end of development.

### 3.4. Evaluation Phase

The Fourth Quadrant, Evaluation Phase, allows lecturers to evaluate their project's output before moving on to the next quadrant. We need to evaluate the system to see whether professors enjoy it and if they have any feedback or suggestions for improvement.

## 4. Results and Discussion

This section presents the implementation and testing that was conducted for Student Attendance System using Facial Recognition.

### 4.1. Implementation

The implementation process starts at the creation of the web application interface. The first interface created was the registration and login page where the user could access their account from. Next, the home page was created followed by student information registration page, lecturer profile page, student profile page and attendance log page. After creating the general interfaces, the next important thing that was developed was the back-end to allow the web application to function. The most important part was the facial recognition panel that is used to detect faces of students and record their attendance in the attendance log.

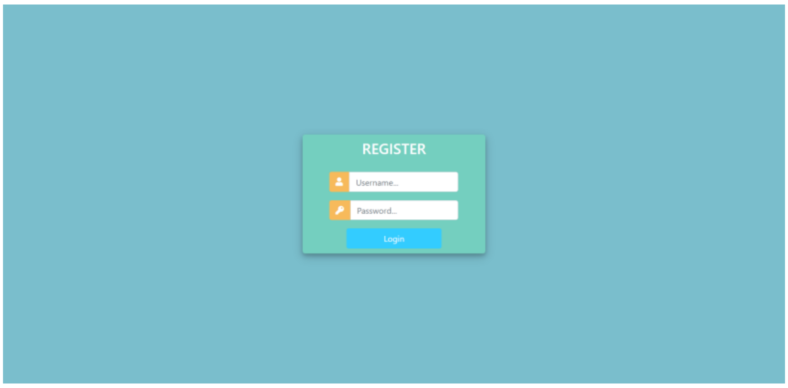


**Figure 6: Registration Page**

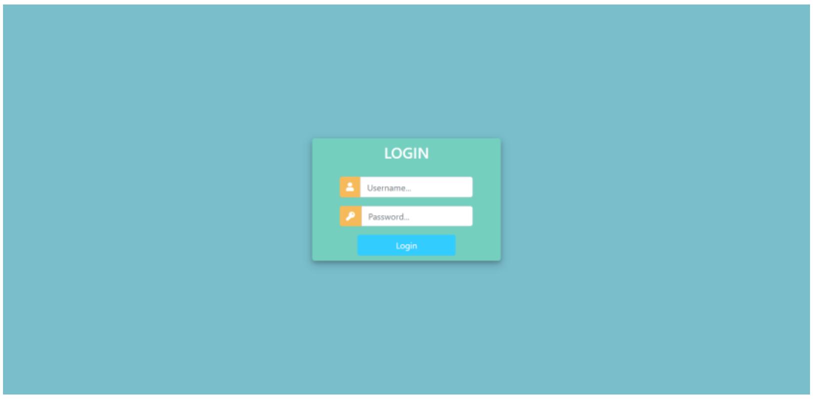
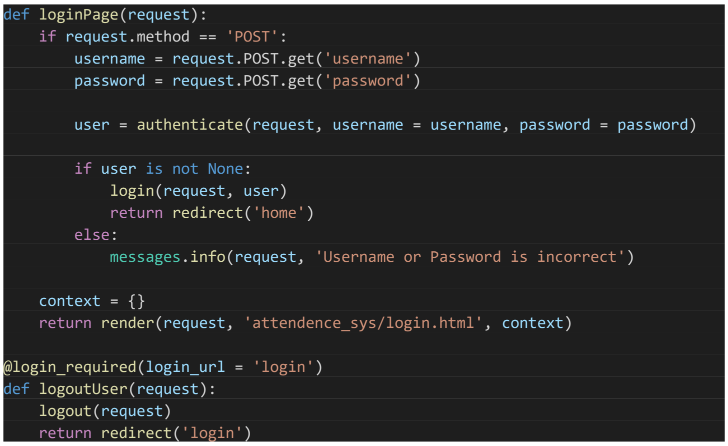


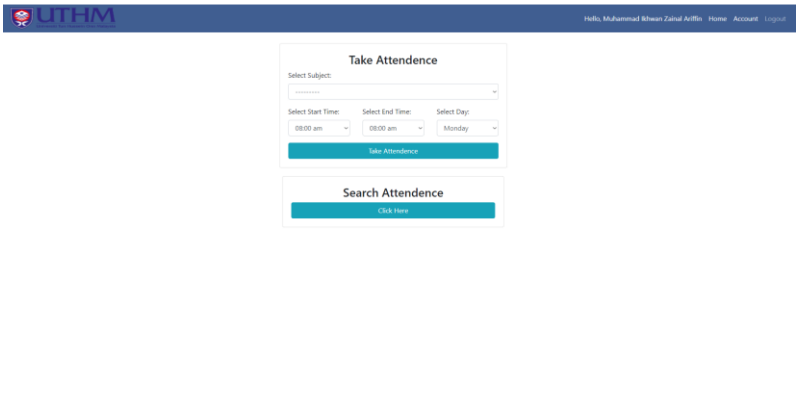
**Figure 7: Login Page**

**Figure 6** and **Figure 7** shows the registration and login page for the system. The lecturer can register and access the web application through this module.



```python
def loginPage(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(request, username = username, password = password)

        if user is not None:
            login(request, user)
            return redirect('home')
        else:
            messages.info(request, 'Username or Password is incorrect')

    context = {}
    return render(request, 'attendence_sys/login.html', context)

@login_required(login_url = 'login')
def logoutUser(request):
    logout(request)
    return redirect('login')
```

Figure 8: Registration, Login and Logout coding segment



Take Attendence

Select Subject: ---------

Select Start Time: 08:00 am

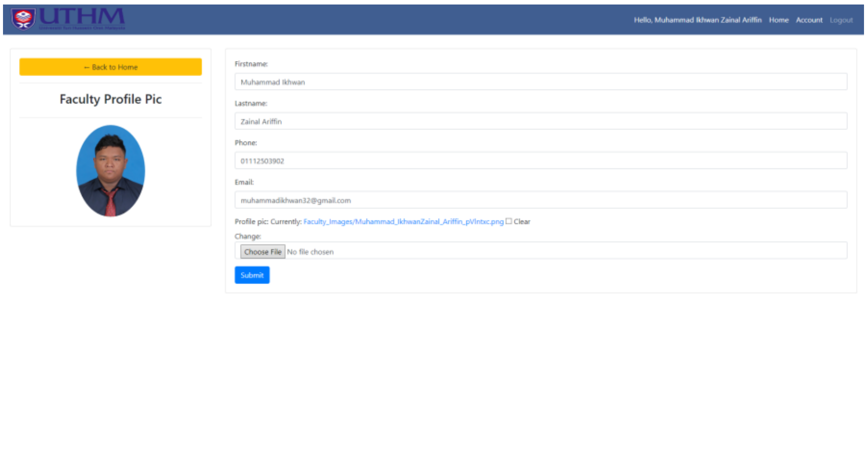
Select End Time: 08:00 am

Select Day: Monday

Search Attendence

Figure 9: Home Page

**Figure 9** shows the home page that appears after successfully logging into the system. Lecturer can choose the subject, class start time, end time, and day of the class which the lecturer wants to take attendance of. The lecturer can also access the attendance of previous classes through this page.



Faculty Profile Pic

Firstname: Muhammad Ikhwan

Lastname: Zainal Ariffin

Phone: 01112503902

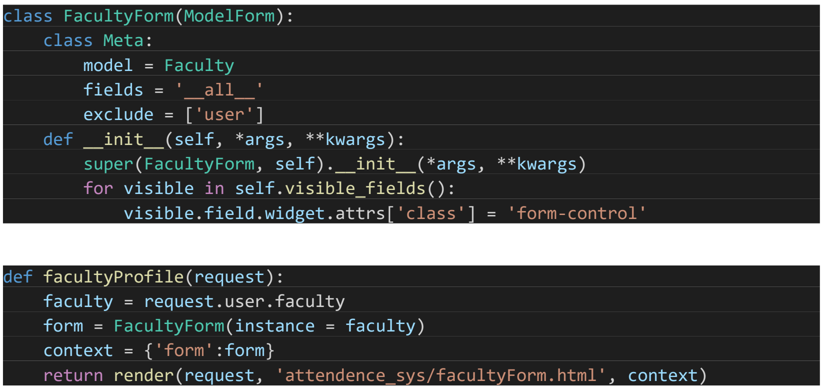
Email: muhammadikhwan32@gmail.com

Profile pic: Currently: Faculty_Images/Muhammad_IkhwanZainal_Ariffin_pVlntxc.png □ Clear

Change: Choose File No file chosen

Figure 10: Lecturer Profile Page

**Figure 10** shows the lecturer profile page. This page lets the lecturer to change their information such as name, phone number, email, and more.



```
class FacultyForm(ModelForm):
    class Meta:
        model = Faculty
        fields = '__all__'
        exclude = ['user']
    def __init__(self, *args, **kwargs):
        super(FacultyForm, self).__init__(*args, **kwargs)
        for visible in self.visible_fields():
            visible.field.widget.attrs['class'] = 'form-control'
```

```
def facultyProfile(request):
    faculty = request.user.faculty
    form = FacultyForm(instance = faculty)
    context = {'form':form}
    return render(request, 'attendence_sys/facultyForm.html', context)
```

**Figure 11: Lecturer Profile coding segment**

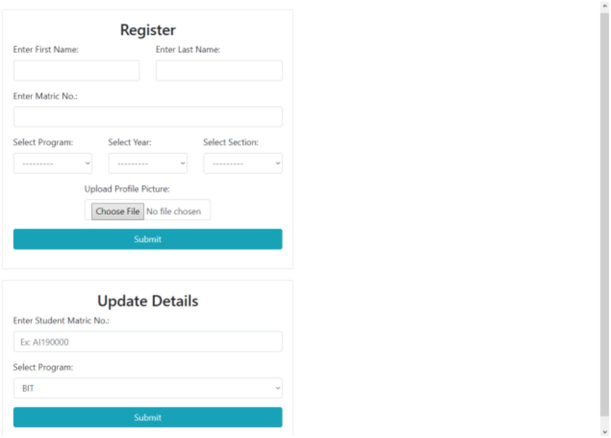
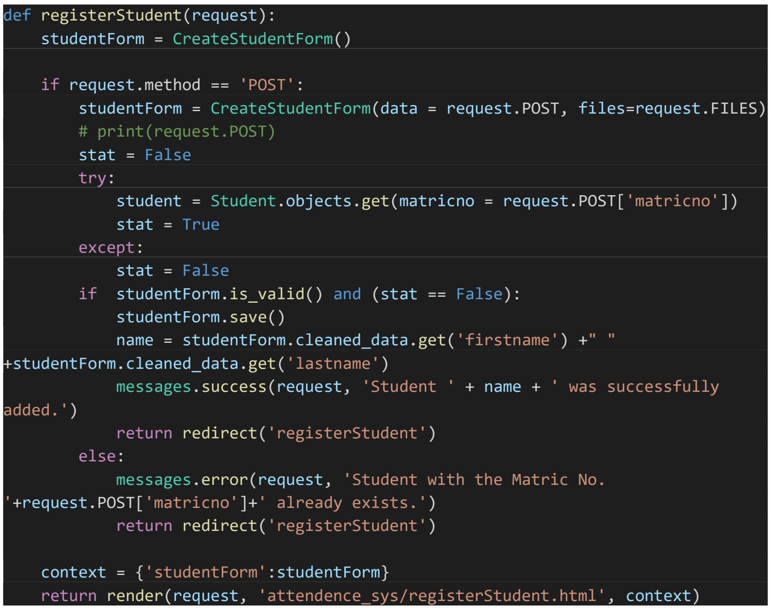


**Figure 12: Student Registration Page**

**Figure 12** shows the student registration page. This page lets the students to register their information such as name, matric number, program, section, year of study, and profile picture. Furthermore, the student can change their information through this page.



```
def registerStudent(request):
    studentForm = CreateStudentForm()

    if request.method == 'POST':
        studentForm = CreateStudentForm(data = request.POST, files=request.FILES)
        # print(request.POST)
        stat = False
        try:
            student = Student.objects.get(matricno = request.POST['matricno'])
            stat = True
        except:
            stat = False
        if  studentForm.is_valid() and (stat == False):
            studentForm.save()
            name = studentForm.cleaned_data.get('firstname') +" "
+studentForm.cleaned_data.get('lastname')
            messages.success(request, 'Student ' + name + ' was successfully
added.')
            return redirect('registerStudent')
        else:
            messages.error(request, 'Student with the Matric No.
'+request.POST['matricno']+' already exists.')
            return redirect('registerStudent')

    context = {'studentForm':studentForm}
    return render(request, 'attendence_sys/registerStudent.html', context)
```

**Figure 13: Student Registration coding segment**

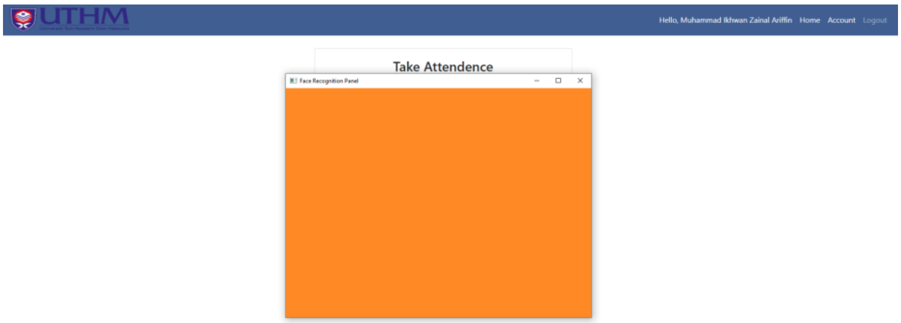
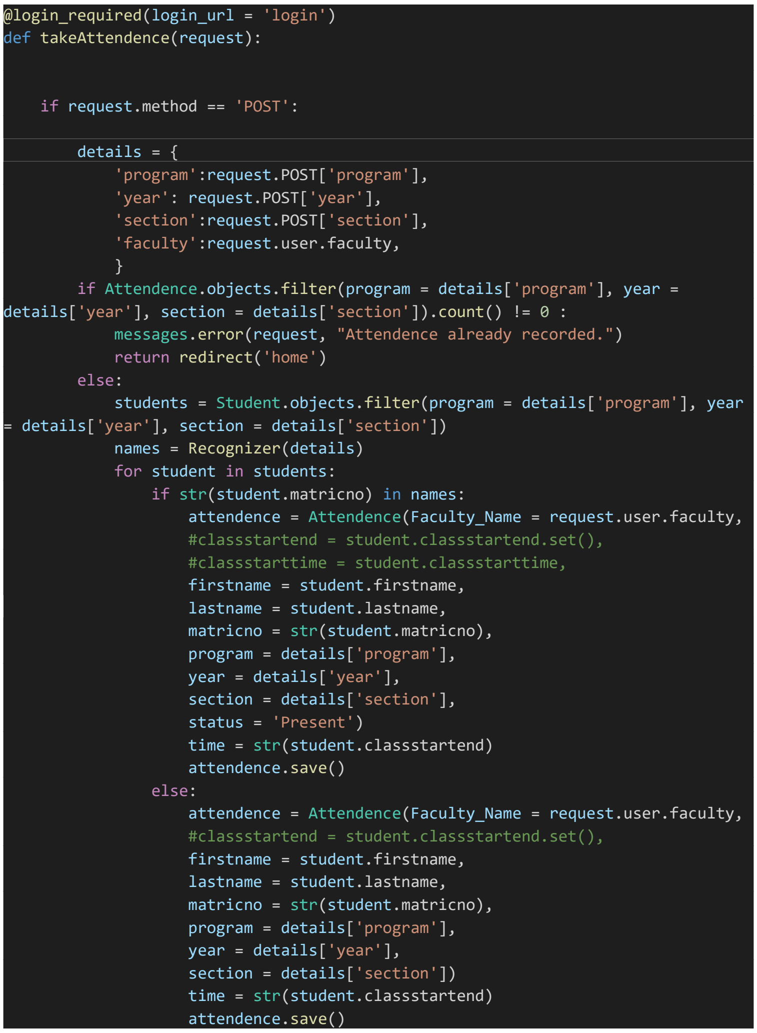
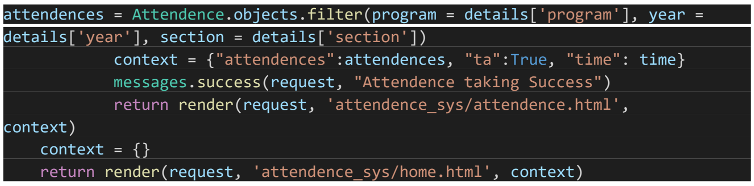


**Figure 14: Facial Recognition Panel**

**Figure 14** shows the facial recognition panel which appear after the lecturer launches the attendance record process. The panel uses the computer's webcam to detect and compare the faces that appear before the webcam.

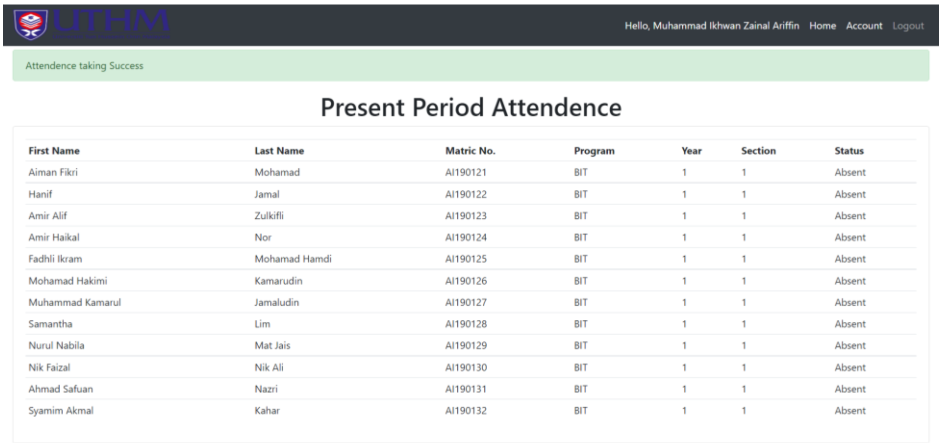


```
@login_required(login_url = 'login')
def takeAttendence(request):


    if request.method == 'POST':

        details = {
            'program':request.POST['program'],
            'year': request.POST['year'],
            'section':request.POST['section'],
            'faculty':request.user.faculty,
            }
        if Attendence.objects.filter(program = details['program'], year =
details['year'], section = details['section']).count() != 0 :
            messages.error(request, "Attendence already recorded.")
            return redirect('home')
        else:
            students = Student.objects.filter(program = details['program'], year
= details['year'], section = details['section'])
            names = Recognizer(details)
            for student in students:
                if str(student.matricno) in names:
                    attendence = Attendence(Faculty_Name = request.user.faculty,
                    #classstartend = student.classstartend.set(),
                    #classstarttime = student.classstarttime,
                    firstname = student.firstname,
                    lastname = student.lastname,
                    matricno = str(student.matricno),
                    program = details['program'],
                    year = details['year'],
                    section = details['section'],
                    status = 'Present')
                    time = str(student.classstartend)
                    attendence.save()
                else:
                    attendence = Attendence(Faculty_Name = request.user.faculty,
                    #classstartend = student.classstartend.set(),
                    firstname = student.firstname,
                    lastname = student.lastname,
                    matricno = str(student.matricno),
                    program = details['program'],
                    year = details['year'],
                    section = details['section'])
                    time = str(student.classstartend)
                    attendence.save()
```



```
attendences = Attendence.objects.filter(program = details['program'], year =
details['year'], section = details['section'])
            context = {"attendences":attendences, "ta":True, "time": time}
            messages.success(request, "Attendence taking Success")
            return render(request, 'attendence_sys/attendence.html',
context)
    context = {}
    return render(request, 'attendence_sys/home.html', context)
```

**Figure 15: Facial Recognition Panel code segment**



Attendence taking Success

Present Period Attendence

| First Name | Last Name | Matric No. | Program | Year | Section | Status |
|---|---|---|---|---|---|---|
| Aiman Fikri | Mohamad | AI190121 | BIT | 1 | 1 | Absent |
| Hanif | Jamal | AI190122 | BIT | 1 | 1 | Absent |
| Amir Alif | Zulkifli | AI190123 | BIT | 1 | 1 | Absent |
| Amir Haikal | Nor | AI190124 | BIT | 1 | 1 | Absent |
| Fadhli Ikram | Mohamad Hamdi | AI190125 | BIT | 1 | 1 | Absent |
| Mohamad Hakimi | Kamarudin | AI190126 | BIT | 1 | 1 | Absent |
| Muhammad Kamarul | Jamaludin | AI190127 | BIT | 1 | 1 | Absent |
| Samantha | Lim | AI190128 | BIT | 1 | 1 | Absent |
| Nurul Nabila | Mat Jais | AI190129 | BIT | 1 | 1 | Absent |
| Nik Faizal | Nik Ali | AI190130 | BIT | 1 | 1 | Absent |
| Ahmad Safuan | Nazri | AI190131 | BIT | 1 | 1 | Absent |
| Syamim Akmal | Kahar | AI190132 | BIT | 1 | 1 | Absent |

**Figure 16: Attendance Log Page**

**Figure 16** shows the attendance log page. This page appears after the lecturer ends the attendance record session. Student names of the class which attendance was taken appear on this page to let the lecturer knows which student attended or was absent.

## 4.2. Testing

After the system has been built, the testing phase begins to investigate the application's performance. Testing was carried out to discover any errors that may occur while employing the Facial Recognition Student Attendance System. Another goal of testing is to see if the application can meet the goal and scope that have been set. **Table 7** shows the summary of the functional testing results.

**Table 7: Functional Test**

| Test | Expected Result | Actual Result |
|---|---|---|
| | Registration and Login Module | |
| User input valid username and password. | System will redirect to the main page of the application while displaying the username. | Pass |
| User input invalid username and password. | System will indicate wrong user login credential is inputted and requires user to input again. | Pass |
| User does not input any login credential during login. | System will indicate which text field is empty. | Pass |
| | Lecturer Module | |
| User initiates the attendance recording process. | Facial recognition panel appears and detects faces of students. | Pass |

| | | |
|---|---|---|
| User attempts to view attendance record. | List of students that attend or are absent with their information. | Pass |
| User changes their profile details. | New details are kept in the database replacing the previous ones. | Pass |
| | Student Module | |
| User registers their information into the system. | Information is kept in the system. | Pass |
| User updates their information. | New details are kept in the database replacing the previous ones. | Pass |

**Table 8: User Acceptance Test**

| No. | Acceptance Requirements | Test Result (Number of people) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Yes | No | 1 | 2 | 3 | 4 | 5 |
| 1. | The system is able to execute smoothly from start to finish. | 20 | 0 | - | - | - | - | - |
| 2. | The system is user friendly and interactive. | - | - | 0 | 0 | 0 | 5 | 15 |
| 3. | The user is able to capture attendance quickly by detecting faces stored in the system. | - | - | 0 | 0 | 0 | 6 | 14 |
| 4. | The system displays the right error message according to the error. | 20 | 0 | - | - | - | - | - |
| 5. | The user is able to manually add or edit students in the attendance list. | 20 | 0 | - | - | - | - | - |
| 6. | The user is able to view attendance list after each recording session. | 20 | 0 | - | - | - | - | - |
| 7. | The system is able to filter the list of attendance according to the information they desire. | 19 | 1 | - | - | - | - | - |

## 5. Conclusion

The system is expected to be able to help teachers or lecturers to record attendance of their students by computerizing the manual method which additionally implement paperless environment concept and making the process faster than usual by doing the process through an online platform.

## Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## References

[1] I. El Naqa, R. Li and M. Murphy, Machine Learning in Radiation Oncology. Cham: Springer International Publishing, 2015.

[2] M. Batta, “Machine Learning Algorithms - A Review ,” Int. J. Sci. Res. (IJ, vol. 9, no. 1, pp. 381-undefined, 2020, doi: 10.21275/ART20203995.

[3] K. Foote, “A Brief History of Machine Learning - DATAVERSITY,” DATAVERSITY, Dec. 03, 2021. https://www.dataversity.net/a-brief-history-of-machine-learning/ (accessed Dec. 25, 2021).

[4] E. Alpaydin, Introduction to machine learning. London, England: The Mit Press, 2020.

[5] N. Nilsson, The Quest for Artificial Intelligence. Cambridge University Press, 2009.

[6] Gigasource, “Facial Recognition Attendance System | Free Download | Fast Custom,” Gigasource, Apr. 28, 2020. https://gigasource.io/face-attendance/ (accessed Dec. 25, 2021).

[7] A. Adel and B. Abdullah, “A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model,” IJCSI Int. J. Comput. Sci. Issues, vol. 12, no. 1, pp. 106–111, 2015, [Online]. Available: https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model.