**CHAPTER 1: INTRODUCTION**

**1.1 Introduction to Internship**

An internship is an opportunity to integrate career related experience into an

undergraduate education by participating in planned, supervised work. The

internship program is designed to provide students engaged in a field experience

with an opportunity to share their insights, to explore the links between students'

academic preparation and their field work, and to assist participants in developing

and carrying out the major research project which will serve to culminate their

internship experience.

Internships are individualized and tailored to the needs and interests of each

student in the program. As part of the internship experience, students are expected

to take an active role in finding an appropriate internship for themselves.

Internship is considered as the practical implementation of the theory education

that provides a chance to use the skills learned in the classroom in a real-world

setting. Internships for professional careers are similar in some ways to

apprenticeships for trade and vocational jobs. The internship is done as a partial

fulfillment of requirements of the Bachelor’s degree in Computer Science and

Information Technology under Tribhuwan University.

Internships are individualized and tailored to the needs and interests of each

student in the program. As part of the internship experience, students are expected

to take an active role in finding an appropriate internship for themselves. Many of the benefits of an Internship Program include:

* developing personally and professionally while gaining confidence and

real-world experience

* meeting and networking with practitioners in one’s area of interest
* mentoring and performance feedback from the site supervisor
* earning academic credit while getting paid.

Internships will provide students the opportunity to test their interest in a

particular career before permanent commitments are made. Some of the objectives

are:

* Provide students the opportunity to relate theory to practice.
* Give students in-service orientation to a career area they may wish to

pursue.

* Allow students the opportunity to work in their area of intended

specialization.

* Enhance student’s understanding of organizational and group processes.
* Enhance student’s awareness of public service obligations.
* Develop an ongoing relationship between the academic and practitioner

community.

* Provide “apprentice” expertise and a new perspective for organizational

operation.

**1.2 Background**

The software market in Nepal is experiencing steady growth. Experts and

professionals involved in this business are taking this growth as a positive aspect

of developing awareness related to information technology in the country. The

trend of using computers is also on the rise since schools, health institutions, clubs

and other social groups in far flung areas of the country have started using

computers for day-to-day activities. Government of Nepal recently showed that

the computer penetration in Nepal is around 3 per cent among which 20 per cent

uses internet. With the rise of mobile applications, the internet users in mobile

have reached up to 67 percent .

The knowledge gained from the academics is generally theory based. The class

room knowledge alone cannot provide the working experience that is required to

work in the real-world setting. The real-world working experience can be

achieved only through internship. The theory education from classrooms along

with the working experience from internship can provide competitive advantage in

building and shaping professional career in today's job market. The strong

academic background alone cannot indicate whether or not the person is capable

of working in a specific work environment.

The objectives of internship were to test theory based knowledge gained

throughout BSc CSIT course and to gain knowledge and experience in IT industry.

**1.3 Objective**

**1.3.1 Objectives of internship program**

The following is a list of some objectives which the internship might fulfill for

both the student and the organization (agency, candidate, interest group, etc.):

1. Provide students the opportunity to relate theory to practice.

2. Give students in-service orientation to a career area they may wish to pursue.

3. Give students opportunity to work in their area of intended specialization.

4. Enhance student’s understanding of organizational and group processes.

5. Enhance student’s awareness of public service obligations.

6. Develop an ongoing relationship between the academic and practitioner

community.

7. Provide “apprentice” expertise and a new perspective for organizational

operations.

**1.3.2 Objectives of Project**

The objectives of the project are as follows:

1. To created an Automated Attendance web application.

2. To develop an easy to use and useful interface for admins to check employee attendance.

3. To be able to use facial recognition to recognize employees to take attendance.

4. To show a POC (Proof of Concept) for the applicability of Amazon Web Services for future applications in the company.

5. To provide information about various employee attendance even to remote admins.

**1.4 Brief Introduction of Industry**

**Time and attendance** systems (TNA) are used to track and monitor when employees start and stop work. A time and attendance system provides many benefits to organizations as it enables an employer to have full control of their employees working hours as it monitors late arrivals, early departures, time taken on breaks and absenteeism. It also helps to control labor costs by reducing over-payments, which are often caused by paying employees for time that are not working, and eliminates transcription error, interpretation error and intentional error. TNA systems are also invaluable for ensuring compliance with labor regulations regarding proof of attendance. All of these benefits provide both employer and employees with confidence in the accuracy of their wage payments all while improving productivity.

**1.4.1 Manual Systems**

Traditionally manual systems were used that rely on highly skilled people laboriously adding up paper cards which have times stamped onto them using a time stamping machine such as the Bundy Clock. Time stamping machines were used for over a century but have since been phased out and replaced with cheaper automated systems which eliminate the need for payroll staff to manually input employee hours.

**1.4.2 Automated Systems**

Modern automated time and attendance systems like ClockIt simply require employees to touch or swipe to identify themselves and record their working hours as they enter or leave the work area. Originally this consisted of using a RFID electronic tag or a barcode badge but these have been replaced by bio-metrics (vein reader, hand geometry, fingerprint, or facial recognition), and touch-screen devices. Modern biometric TNA systems offer additional benefits over traditional manual systems which include

* Faster processing of employees as attendance can be recorded with just one touch or a quick scan
* Fraud prevention by eliminating duplicate and fake registration
* Saves time as attendance can either be integrated directly with your payroll system or it can produce a report that can be downloaded or printed
* Improves punctuality and reduces long breaks and absenteeism

**1.5 Brief Introduction of Organization**

**1.5.1 About the Organization**

Founded in 2015, Ashleesha International is a premier software development company driven towards providing the best of the technology era through solutions locally and globally targeted mainly for the UK client. Ashleesha International delivers technology-driven business solutions that meet the strategic objectives of our clients and create unmatched business value for clients through a combination of process, excellence, quality frameworks and service delivery innovation.

Ashleesha International is an ever-growing community of passionate and brilliant people. Our exceptional team has every solution to our clients' biggest and most complicated problems. We incorporate best practices like collaboration, feedback, transparency identifying requirements for a complete solution. Our services are offered both on and offshore, and delivered with pride and passion. Our customers are diverse, spread across the enterprise, industry and media segments of all sizes ranging from startups to large enterprises who realize that they need a professional

solution to generate revenue streams, establish communication channels or

streamline business operations.

**1.5.2 Organization Rationale**

**Vision**

We provide world class services in the field of Software Development through our

knowledge and skill, mainly for UK clients.

.

**Mission**

We generate technically confident and professionally motivated team to

deliver tasks in time. We simplify, standardize and automate the

organizations technically. Analyze, plan, develop, deploy and maintain are

the major steps for any new product or service we create. Our main

strategy is to combine best people and best equipment, to create the best

technology and product.

**Our Values:**

Honesty, Trust, Freedom, Innovation, Integrity, Team spirit and Fun are

some values influence the way we meet client needs.

**1.5.3 System Architecture**

Previous Attendance System in use was primitive. It was done using pen and paper. Calculating the absent and present days had to be done by a human manually and that was a major time waster and was tedious. It was also more vulnerable to manipulation and forgery.

Hence, our Automated Attendance System has been developed as a better replacement to the traditional system of pen and paper.

The advantages that are in current system are cited below:

Very Informative

Interactive to the administrators

Prevents forgery and manipulation

Quick reporting and very fast calculation of absentees and present employees

Use of Facial Recognition technology improves employee morale and company image

**1.5.4 Contact Details**

Ashleesha International

Mulpani, Kathmandu, Nepal

Phone: 9841378626

Email: prashantdhungel@gmail.com

Slack: phuyalgroup.slack.com

**CHAPTER 2: ANALYSIS OF ACTIVITIES DONE**

**2.1 Activities Done**

**2.1.1 Organization selection**

The four year degree of BSc.CSIT allows us to attain knowledge on various

aspects of Information Technology. At the same time the internship is the one of

the major highlight of the program to expose the students to the professional

world. Among the various criteria and sectors provided to us in internship

prerequisite statement, web based software development was chosen. Various

organizations were shortlisted and approached out of which the organization with

the best lucrative offer and environment was selected. I selected Ashleesha

International Pvt. Ltd. as my intern organization.

**2.1.2 Placement**

During the internship tenure, I was provided with work space at the organization,

not only that I was treated as trainee staff with the access to the various information

of the organization, its organizational resources, information and equipments.

**2.1.3 Duration**

Start date: November 9, 2017

Total duration: 3 months

Position: Intern for three months. Trainee Software Developer

Supervisor: Mr. Prashant Dhungel (Project Manager)

Office hour: 10:00 A.M – 6.00 P.M.

**2.1.4 Roles and Responsibilities**

During the internship period at Ashleesha International Pvt. Ltd., the target was to

deliver a fully fledge web based “Automated Attendance Application” by using different tools and frameworks such as Python, ReactJS, HTML/CSS, p5js, Amazon Web Services, AWS Lambda, AWS DynamoDB, AWS Rekognition, AWS S3, Trello, etc. Hence, to meet that objective, this internship required the extensive preliminary studies

about the core Python, Amazon Web Services, HTML/CSS, Trello before actually analyzing the actual requirement of the system. The study was required not only to

understand the subject under study but also to realize the solutions to the existing

problems and implementing the findings from the study was another bigger

challenge.

**2.2 Literature Review**

During internship period author went through various papers, journals and

literature that were published. Author researched and analyzed on different topic

and came to the conclusion with different title. **Time and attendance** systems (TNA) are used to track and monitor when employees start and stop work. A time and attendance system provides many benefits to organizations as it enables an employer to have full control of their employees working hours as it monitors late arrivals, early departures, time taken on breaks and absenteeism. It also helps to control labor costs by reducing over-payments, which are often caused by paying employees for time that are not working, and eliminates transcription error, interpretation error and intentional error. TNA systems are also invaluable for ensuring compliance with labor regulations regarding proof of attendance. All of these benefits provide both employer and employees with confidence in the accuracy of their wage payments all while improving productivity.

However, a prerequisite to set up a successful new software company can be an automated attendance web portal. An automated system is the first thing the employee interacts with when they reach the company premise, so it has to be cool, attractive and user friendly to improve the moral of the employee when they enter the company.

These days there is no dearth of companies providing automated attendance application services. It is highly necessary for a progressive minded software company. Hence from

the project we have tried to provide a good and well-designed custom web design

ensuring customization, flexibility, robustness, compatibility, attractive visual

appearance, easy access, etc.

Being a reputed and experienced software development company we tried to ensure that we get a thoroughly professional and visually attractive attendance application.

This is imperative for a company’s smooth functionality. Also, it should have a clean and error free structure. This will prevent the glitches. An erroneous system can hamper the functionality and can affect its business.

An Automated Attendance System helps to establish credibility as a modern software company. There are actually still quite a few small businesses that

don't have an automated attendance and without one this is exactly what they will remain.

If a company doesn’t have an automated attendance system, then the employees are going to assume that company is a small time company that does not take their business seriously and once these reputations are established it is going to be hard to make the employees give their 100% effort.

Not too long ago, if a person started a company, he/she might use an attendance register on paper in the hopes of taking the attendance for the company in the

area. But, being a software company has completely changed that practice. The standard of expectation is higher in a software company and the employees of a software company would prefer to use an Automated Attendance Application. An Automated Attendance gives the company a better expectation and outlook to the employees.

Amazon Web Services could be considered as a good and a quite quick way for small

companies to build their products and even use the most advanced “Machine Learning Services” directly like Amazon’s Rekognition API.

Moreover Cloud Computing Services of Amazon Web Services, with all its advantages, brings a significant improvement in productivity, which is likely to improve the image and the notoriety of a product, a brand or a company.

**2.3 Specific Problem Analysis**

**2.3.1 Understanding the existing website**

Before the new system is developed we analyze the previous system being

operated. The previous system has its own advantages and some lacking as well.

So we need to understand the previous system. Then in the new system we should

be able to remove previous system faults.

One of the most important tasks is to recognize the real problem of the preinstalled

system. The analysis has to spend hours and days for understanding the

fault in the system. This fault could have however overcome if the Preliminary

Investigation before installing the system was properly done. This is the first stage

of the development of the system. In this stage the analyst makes a survey by

gathering all the available information needed for the system elements and

allocation of the requirements to the software.

In this step what we did was studied the existing Attendance System. The

current system was using was simple register and just attendance was counted by the employees themselves. It would be tedious and time consuming to find your index and do the attendance. It was also difficult and time consuming for the office accountant to do the counting of employee attendance for every employee in the company. Also, there was no section where the employee could explain about what they did yesterday and what they were planning to do today. Another problem was everyone had to take turns to do the attendance as there was only one physical book where the attendance was done in. There was also a slight feeling that the employees of a software company felt somewhat demoralized when doing the attendance by pen and paper and doing this everyday reduced the moral and expectation standards of the employees. Thus, there was a big scope in the possible enhancements we could make upon the Attendance System.

Understanding the previous system was very helpful for us, as we can see what

the present scenario is and where we have to work further to make this system

better. We can see the flaws and study the lesson about what should not be done.

We can also list the present advantages of the system and carry it on to the next

architecture as well.

**2.3.2 Development of Project Goals**

We learnt the recent trends in the attendance application and studied how to

implement them in our system technologically. Also having the researched

undertaken about the user need was done so that the system can better address the

user needs. And also special considerations were made to make the system user

friendly by studying about the possible user interfaces that the system can have.

Also the portal is supposed to have every transaction carried on through a web application, so fraud detection was a crucial need. An employee should only be able to check their attendance from the office and shouldn’t be able to do it from their home. Hence the measures and mechanisms for maintaining security

were also researched on and reliable attendance system was to be implemented.

Through the Automated Attendance Application, we intend to provide a better platform for the company to take the attendance of the employees. We shortlisted the features of the project to include the following:

A web interface for the employees of the company. The Employee takes a picture through the web cam and the system uses that to check their attendance by using Facial Recognition Technology and also update their attendance in the database.

In addition it can be helpful to let the employee provide more information like what the employee did yesterday and what they’re planning to do today.

Make the admins be able to check the employees’ attendance for the entire duration from days, months to even years.

Make the admins be able to view the attendance from a remote site since the data is stored in a web database.

To shorten the attendance taking time for the employees.

To save the time required to count up all the attendance of all the employees.

To improve the morale and expectation standards of the employees since they’re seeing the face recognition technology being used every time they enter the company building.

**CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

System development is the process of understanding the development of any system which is carried out using two different components: System Analysis and System Design

**3.1 Project Management Plan**.

## 3.1.1 System Analysis

Systems analysis is a problem solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. System analysis helps to understand the proposed system architecture, working and goals.

### 3.1.1Feasibility Study

A feasibility study is an analysis of the ability to complete a project successfully, taking into account legal, economic, technological, scheduling and other factors. Rather than just diving into a project and hoping for the best, a feasibility study allows project managers to investigate the possible negative and positive outcomes of a project before investing too much time and money.

#### 3.1.1.1Technical Feasibility

The technical feasibility is an evaluation of the hardware and software and how it meets the need of the proposed system.

The system is developed using Amazon Web Services as the main development tool according to client’s specification. DynamoDB, a cloud real time database is chosen to be the DBMS of the system, due to its compatibility with AWS. And, ReactJS is used for the front end development.

* Availability of the chosen technologies: All the chosen technologies were made available by the company. These technologies are mostly proprietary (especially the Face Recognition Technology provided by Amazon Web Services Rekognition) and requires registering and purchasing of the service on a pay-as-you-go Model.
* Technical Expertise: The Company didn’t have technical expertise in above chosen technologies. But, since the project wasn’t a large scale project and a Proof of Concept project, so building the project would require some research and practice of developing python powered applications in Amazon Web Services.

#### 3.1.1.2 Legal Feasibility

Legal feasibility assessment determines whether the proposed system conflicts with legal requirements; for instance, a data processing system must comply with the local data protection regulations.

#### 3.1.1.3 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives.

The primary focus of the project is to provide a modern interface that uses Face Recognition Technology to quickly take the attendance of the employees. The employee only has to enter their name and click on a button to start the facial recognition. The administrators have to login into the application first. And then they can view all the employee attendance records for the entire duration and also get quick reports. Sessions, hash values, expiry time and Tokens were used to implement the login system.

#### 3.1.1.4 Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. The following table represents the schedule of the activities that are needed to be performed.

Table 3.1: Project Schedule

|  |  |
| --- | --- |
| **Tasks** | **Duration (days)** |
| Requirements Gathering | 14 |
| Feasibility Study | 7 |
| Design | 14 |
| Development | 21 |
| Testing | 7 |
| Demo | 1 |

## 3.1.2 System Design

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development.

### 3.1.2.1 Functional Partitioning of the System

The functional component of system is partitioned into three different groups as an Administrator and a User.

Check overall attendance pattern of each employee overt

Administrator

View the list of present employees

Add/ Edit/ View/ Delete Users

View the list of absent employees

**Figure 3.1: Functional Partitioning of administrator**

User

Take attendance with Face Image

Add additional information for the day

View status for the day

Figure 3.2: Functional Partitioning of user

System

Use Face Recognition to take attendance of the Employee

Login Functionality for Admin

Figure 3.3: Functional Partitioning of System

### 3.1.2.1 Functional Description

**Administrator**

The functions of administrator are:

* Add/Edit/View/Delete Users
* View the list of absent employees
* View the list of present employees
* View overall attendance of all employees
* Upload the face image to be compared for Face Recognition

**Users**

The functions of users are:

* Sign Up
* Take attendance by taking a picture of their face
* View attendance status for the day
* Add additional information for the day. E.g. the task performed yesterday and the task to be performed today

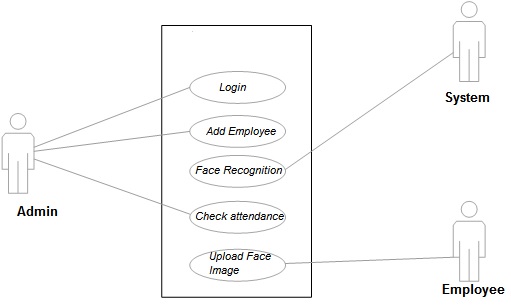
**System**

* Login Functionality for admin
* Use Facial Recognition Technology to take attendance of the Employee. Only takes the attendance if the face is over 80% similar to the one stored in S3 bucket.

**3.1.2.3 Use Case Diagram**

A use case diagram is a type of behavioral diagram defined by the Unified Modeling Language (UML) whose aim is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. It is used to identify the primary elements (actors) and processes (use cases) that form the system. Use cases are used during the analysis phase of a project to identify and partition system functionality.

Figure below describes the use case diagram for Automated Attendance System.



**Figure 3.4: Use-case Diagram for Automated Attendance System**

**Use Cases**

**Use-Case #1: Taking Attendance Process**

**Actor:** User

**Pre-condition:** User should already have a recent reference image of their face stored in the S3 bucket beforehand. The reference image shouldn’t be very different from the one being uploaded. For example, if you had a beard before and now you don’t, maybe you’d want to ask the admin to use a new reference image where you don’t have a beard.

**Success:** The name of the employee is filled and an image with the face in the proper orientation is taken from the web cam, and the records are stored in the database. The final success message is shown on the screen with the similarity % in the screen or console.

**Failure:** The employee is not oriented properly for the face image or has recently had a change in their facial image. E.g: glasses, beard, object obstructing, or wrong orientation. The attendance is not taken and an error message is shown like “No Face Detected or only less than 80% similarity”.

**Use-Case #2: Admin Login Process**

**Pre-condition:** Admin should not be logged in.

**Success:** The correct credentials are filled in the login form, and the records of the employees are fetched from the database. Logged In message is shown.

**Failure:** When the admin enters wrong username or password.

**Use-Case #3: Face Recognition Process**

**Pre-condition:** Registration and login process must be followed. The facial recognition should only be enabled after the admin has given the permission.

**Success:** The Similarity Score is shown, greater than 80% and the correct data is filled in the employee table in the DynamoDB table in the backend.

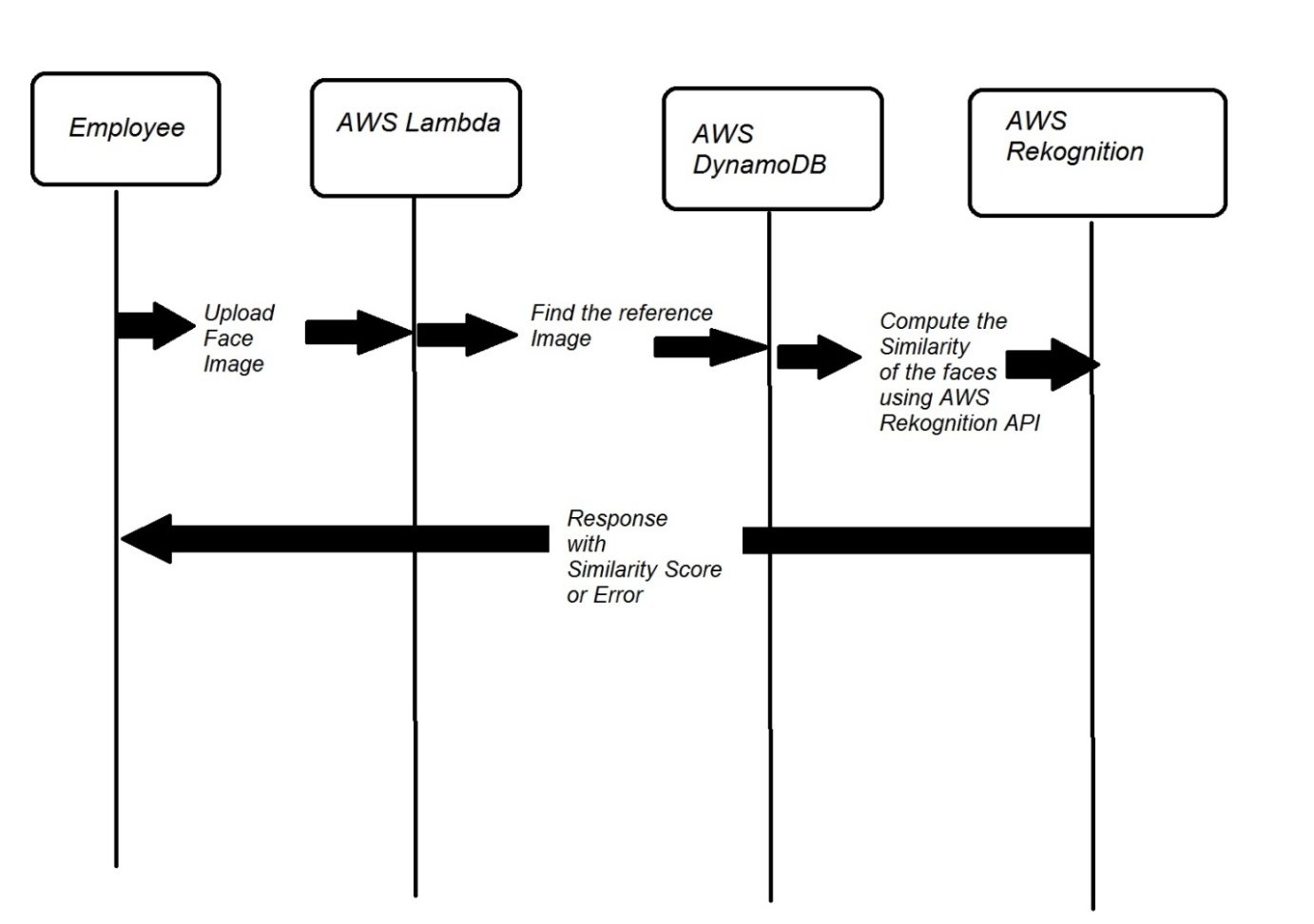
**Failure:** Employees are shown an error message on the problem and no changes are made in the database table. They are redirected back to the Face Image Upload Page.

**3.1.2.4 Sequence Diagram**

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



**Figure 3.5: Sequence Diagram for Employee Interaction in Automated Attendance**

In above mentioned sequence diagram at very first the employee has to go take a picture of their face where they may also have to provide their name. Here the registration of the employee is done by the administrator and also the reference image is also uploaded by the administrator. The image is then base64 encoded so that it can be transferred through http protocol and it is received by an API Gateway where it sends the data to a lambda function “employee attendance” which is written in Python 3. The image is then compared with the reference image previously stored in the S3 bucket or the DynamoDB database. The reference image is found and then both the reference and the uploaded face image of the employee are sent to the AWS Rekognition service. The AWS Rekognition service is a machine learning service with many deep learning applications. Here, the service being used is Face Recognition. The two face images are taken by the AWS Rekognition Service and then a similarity score is returned from the service. This is informed to the AWS Lambda function that then compares it to the threshold value (currently 80%) and then if this threshold is exceeded, then it makes necessary updates in the database. Otherwise, the error message is sent to the Employee back to the web interface.

**CHAPTER 4: IMPLEMENTATION STRATEGIES**

A system must be implemented for its use. The system was developed mainly

through writing codes but some of the components were taken from pre-existing

systems. Specially, Python 3 was chosen as the programming language as per the

requirements of the client. DynamoDB was used as the database for the system also as

per the specification of the organization. ReactJS and p5js were used for the front end application. Amazon Web Services were used for the static hosting, API Gateway, AWS Rekognition Service for Facial Recognition, [28].

**4.1 System Testing**

System testing of software or hardware is testing conducted on a complete,

integrated system to evaluate the system's compliance with its specified

requirements.

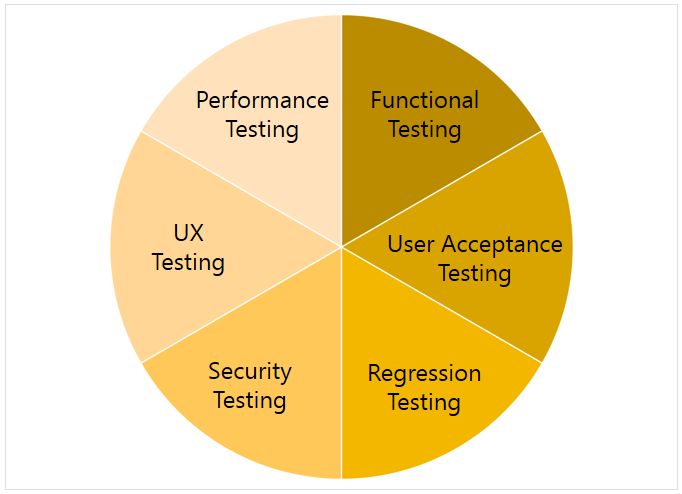


Figure 4.1. Type of testing

**4.1.1 Unit Testing**

The unit tests were carried out on each individual component as they were being

developed. Since Visual Studios provided a good built-in feature for this it was

carried out easily during the component development itself [29].

**4.1.2 Module Testing**

Though each of the components developed may give positive result but when

integrate with other components of the system new properties may emerge

through interfacing. Each of the UI elements like the Grids, Menus, and

Pagination are developed as module. For instance the Grids in isolation may work

perfectly fine but when working with the other components in a page may behave

abnormally.

**4.1.3 Integrated Testing**

Integrating of different modules comprise a system. The major issue to be

addressed while designing the IMS was the complexity in assigning the user roles

and following the approval process that interacted with number of tables and

caused lot of changes. Once all the required modules were developed and

integrated, the system was tested to ensure that the goal had been attained and

rectification was done per the requirement.

**4.1.4 Users Testing and Discussion**

The codes were regularly sent to the Quality Assurance Team for testing each

prototype being developed and regular discussions were held to access the

situation and have any changes incorporated in the system immediately.

**4.1.5 Test Cases**

The test cases were written together in the AWS lambda function. With AWS lambda, it was possible to create multiple test cases of what the result should look like provided from the lambda function for certain events. This was essentially “Unit Testing”.

**CHAPTER 5: RESULT ANALYSIS**

**5.1 Result**

Automated Attendance System can be a powerful tool to monitor the attendance of the employees in the company. And by developing a Automated Attendance System, the main aim was to create a web application that improves the productivity and morale of both the employees and the administrator who can now remotely view the attendance of each employee and get quick and fast attendance reports. The project was also done to show the Proof of Concept of using AWS Services to quickly build useful web applications for the company using the latest technologies like “Face Recognition Technology”.

Hence, developing a web based automated attendance system is aimed to serve this purpose. The web application that author has developed could be solution to overcome the problem of improving productivity of the company through web based application.

This application has following features:-

1. Allow administrator to add and delete employees

2. Employees are able to quickly take their attendance using just their name and face.

3. A well-organized dashboard on the admin panel.

4. Admins can keep track of present and absent employees.

5. Admins can quickly find the total present days of an employee.

6. The System uses Facial Recognition Technology provided by the Amazon Web Service called Rekognition.

7. Automatic insertion of attendance record for the employee when the similarity score exceeds a predefined Threshold.

In this project I was assigned to develop the aws module for the Automated Attendance App which gives the authentication for the admins, updates the records based on the similarity of the reference and uploaded images, uses Rekognition API from Amazon, provide the api for the data involving the employees the admin would need, etc.

**5.1.1 Screenshots**

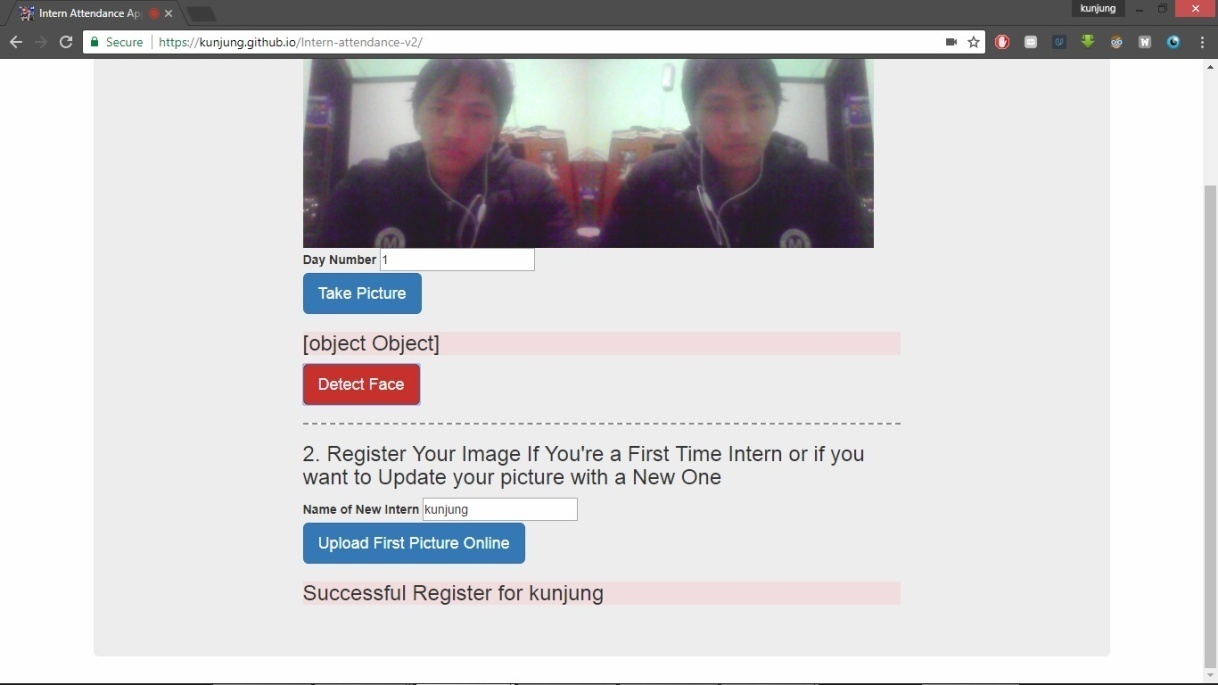


Figure 5.1. Screenshot of Automated Attendance Reference Face Image Registering

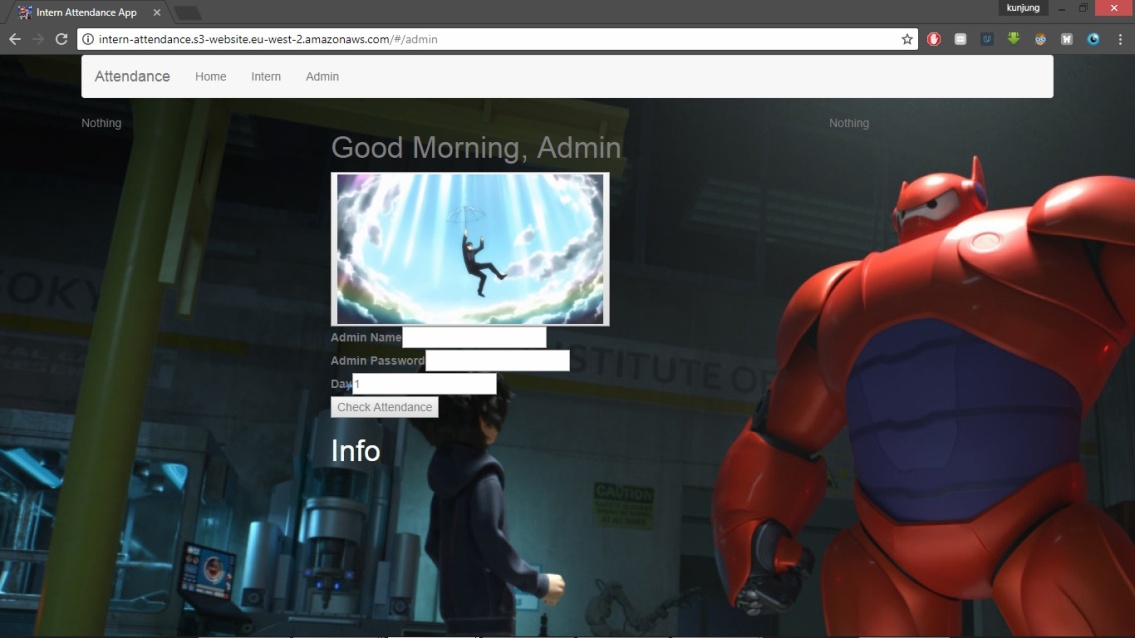


Figure 5.2. Admin Panel before Login

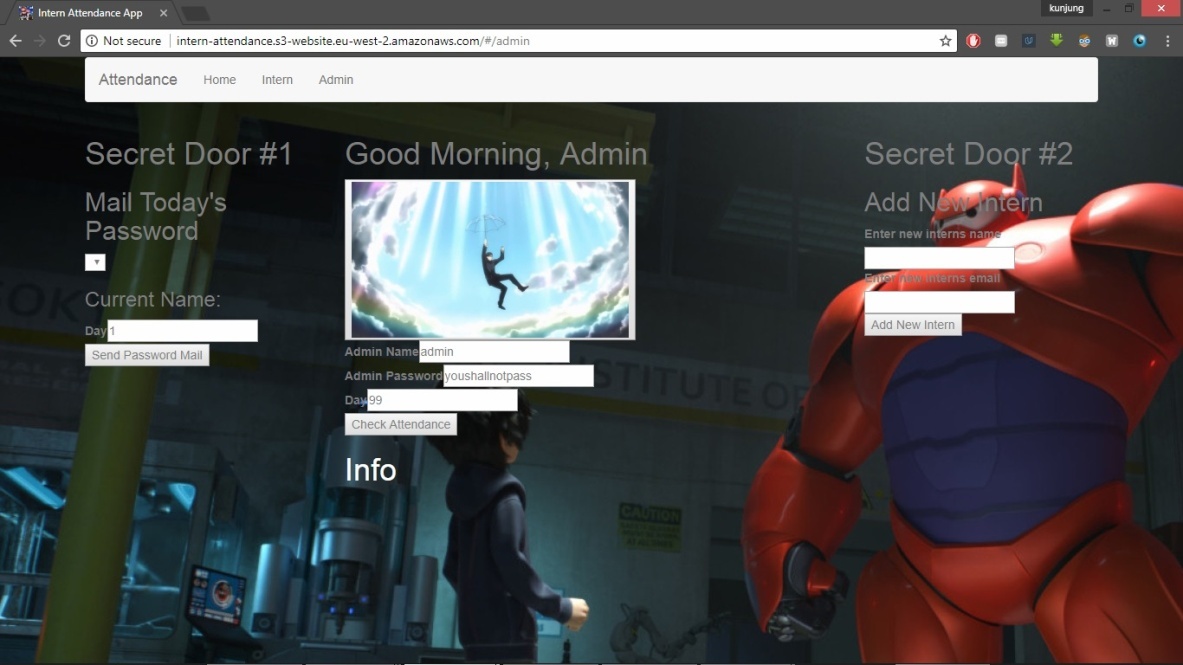


Figure 5.3. Admin Panel After Login

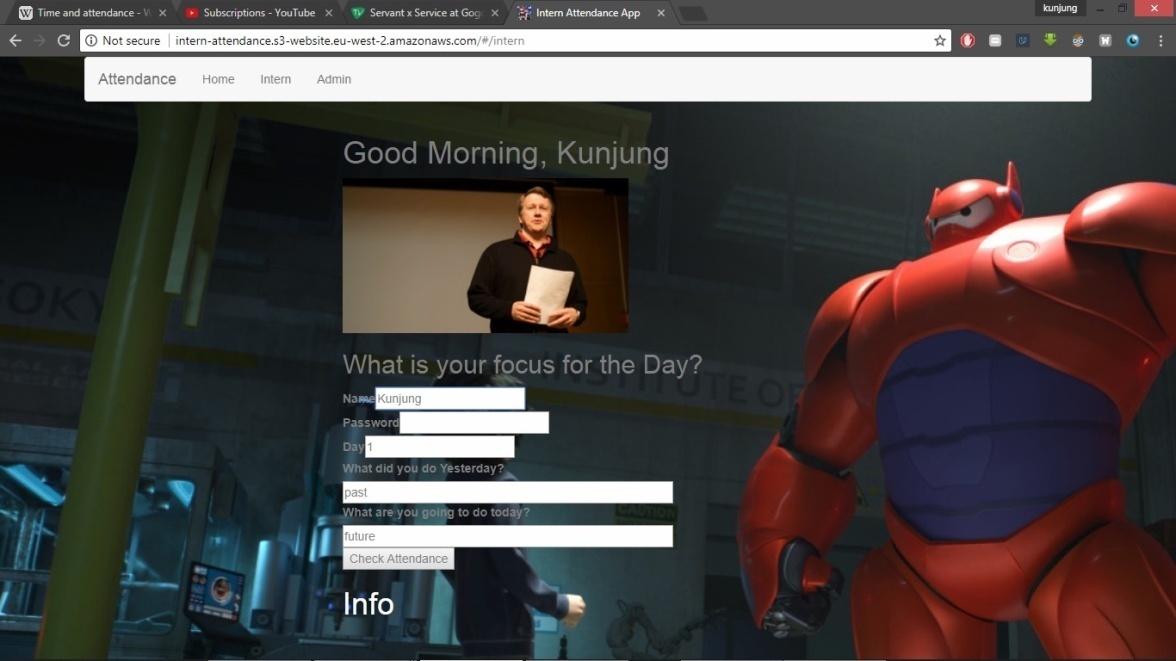


Figure 5.4. Employee Dashboard

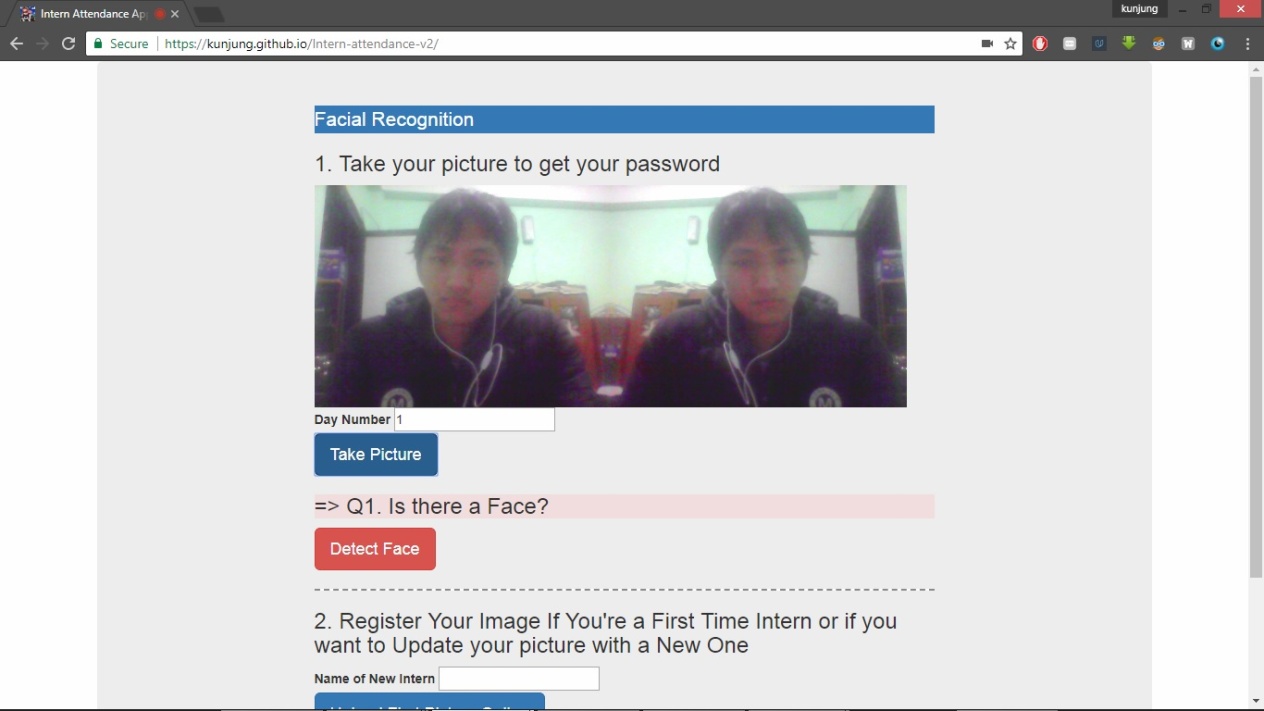


Figure 5.5. Face Recognition Dashboard

**5.2 Critical Analysis**

On the basis of research various requirements of the system were extracted,

analyzed and finalized. The finalized requirements were used to develop a web application that was more efficient for checking through all the reference images in the S3 bucket. The development was carried out systematically through various diagrams and designs including architectural, modular, procedural, data and interface design. Different

development tools were used to develop the application.

The overall system was developed in a team using Trello. Trello is a web-based project management application originally made by Fog Creek Software in 2011.

Trello has a variety of work and personal uses including real estate management, software project management, school bulletin boards, lesson planning, and law office case management. A rich API as well as email-in capability enables integration with enterprise systems, or with cloud-based integration services. With Trello, you can get rather creative. But, it may require some experimentation to figure out how to best use it for your team and the workload you manage.

Due to a simple list of options available it won’t take long for the administrator to actually understand the admin interface as well as the employee to understand the employee interface.

**5.3 Limitations of the system**

Upon successful development of first phase of the Automated Attendance development,

focused on the user interactivity and providing more web solution for the employee. The second phase of this system is under the planning and

this time the systems features is going to be extended so that it can accommodate

the login system for the employees as well where today only admin have login systems implemented for them. Then, the employee will be able to view their attendance record for themselves as well. Some of the Limitations on the existing system are:

Login System restricted to Admin Only

The web app needs to be given Admin Privileges before running Facial Recognition.

Therefore, the admin has to be present at the site, at least initially to allow facial recognition for the remaining employees.

Authentication and Authorization have been implemented by the author himself instead of using a pre-built Authentication Service provided by Amazon Web Services.

Only Uses a Single Reference Image for the Face Image Comparison. Could be improved by having multiple reference images and comparing from any of them and returning the highest similarity score so that the employee not need to ask the admin to upload a new reference image every time their facial features change significantly.

Attendance report not integrated to salary and performance calculation

Attendance Patterns of the Employee might be a useful feature to include in the future. But, it’s currently not present.

**5.4 Recommendation to the Organization**

Ashleesha International Pvt. Ltd. has furnished me with all the necessary help and

support which is really worth mentioning. However, there are few of the things to

be recommended. The various equipment’s necessary for the internship work

should be made easily available to the interns. As the interns have only

theoretical knowledge, the organization should understand that they need time to

time guidance and supervision. As the caterpillar struggles to change into butterfly

so are the internees who are still students struggling to change into professionals.

Hence, the internees may be treated as the professional workers of the

organization, but they are still students learning to gain practical knowledge as

they lack real world knowledge and practical implementation of the theoretical

knowledge that they have gained in the academic institute.

**5.5 Recommendation to the Internship Program**

The university has put great effort to include various courses that would help in

the real world working environment. However, still lots of efforts are to be made

because the current courses and knowledge gained in the college is still not

sufficient enough to work in the practical environment.

The university must execute some research and studies to select the courses that

fulfill the current market requirements that would help the students to qualify

themselves in the practical implications as well.

**CHAPTER 6: CONCLUSION**

I got great opportunity to get enrolled in the development of system which was

solely aim to manage different components and tasks efficiently and effectively.

The system has been successfully developed successfully in its first phase and the

second stage development is underway. The organization has been continuously

been working to promote the use and development of in house software. The experience of working along with teams to build the final product was a great learning experience. Moreover, it presented an opportunity to work in a

development platform – Python that I have always adored. For the first two weeks, I

was overwhelmed by the depth in the cloud computing services provided by “Amazon Web Services” required to carry out the system development work. Although I had a workable knowledge of core Python, the building of custom components, coding standards… and above all new environment to work at; all of which I had to get familiar and that to put to work very soon! I got the chance to get acquainted with professionals who were building products for the International Market, especially in UK. I had

gained much knowledge and skill to understand, dissect, plan and solve the real world problem.

But as the days progressed things got easier and difficult during certain hurdles (of a completely new and strange idea or new platform) and everyday was an amazing learning

experience. I still remember the difficulty that I had building my first custom

component – which took me nearly about a week. But once it was through there

was no looking back and one of the most memorable moments was when I was

able to get a complete module working all on my own. Another important thing

that I learnt was using a “Trello Board”, “Stand Up Meetings” under the principle of Agile Method. The important knowledge which should be kept in mind while working in a team environment for real world project and the ability to merge various services provided by AWS to create a full fledged application were learned through this internship project.

The major achievement of the internship program for me is the:

Learning and exploration of Amazon Web Services like Lambda, DynamoDB, S3, API Gateway, Amazon Rekognition, etc.

Learning to implement a Login System using tokens, hashing, expiry date, localStorage, etc in Python 3.

Working in a Team and deciding on the task each team member should complete as well as resolving conflicts.

Apart from that I, also learnt various other lessons such as:

I learnt about the various changing trends of the software development industry mainly regarding Cloud Computing.

I have gained the knowledge of the practical environment of the

organization that I have been the part of.

I have learnt the organizational discipline and time management.

The exposure to the practical environment has increased experience and

confidence to deal with various organizational communications with

customer.

I have learnt the presentation and demonstrations skills as well.

Finally, this internship program has improved my skill level to communicate with the team members regarding any issues that arise during the project development.

**CHAPTER 7: APPENDIX**

**7.1 Source Code**

**Face Recognition Template:**

<html>

<head>

<link rel="icon" href="favicon.ico" type="image/x-icon" />

<!-- Latest compiled and minified CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.16/p5.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.5.16/addons/p5.dom.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.17.1/axios.min.js"></script>

<title>Intern Attendance App</title>

</head>

<body>

<div class="container">

<div class="jumbotron">

<div class="row">

<div class="col-md-8 col-md-offset-2">

<p class="bg-primary">Facial Recognition</p>

<h3>1. Take your picture to get your password</h3>

<form>

<div id="container"></div>

<label>Day Number</label>

<input type="text" id="day" name="day" value="1"/>

<br/>

<button id="takepicture" type="button" class="btn btn-primary btn-lg">Take Picture</button>

</form>

<h3 class="bg-danger" id="hasFace">=> Q1. Is there a Face?</h3>

<button id="detectface" type="button" class="btn btn-danger btn-lg">Detect Face</button>

<hr style="background-color: #fff; border-top: 2px dashed #8c8b8b;">

<h3>2. Register Your Image If You're a First Time Intern or if you want to Update your picture with a New One</h3>

<form>

<label>Name of New Intern</label>

<input type="text" name="intern\_name" id="intern\_name" required/>

<br/>

<button id="firstpicture" type="button" class="btn btn-primary btn-lg">Upload First Picture Online</button>

</form>

<h3 class="bg-danger" id="hasUploaded">=> Q2. Has Uploaded or not?</h3>

</div>

</div>

</div>

</div>

<script type="text/javascript" src="app.js"></script>

</body>

</html>

**Face Recognition Portal:**

let FACE\_RECOGNITION\_URL = 'https://acqdzcwzj4.execute-api.eu-west-2.amazonaws.com/prod/detectface';

let FACE\_UPLOAD\_URL = 'https://acqdzcwzj4.execute-api.eu-west-2.amazonaws.com/prod/uploadface';

let capture;

let container;

let takePictureButton;

let detectFaceButton;

let firstPictureButton;

let hasFace;

let hasUploaded;

let width = 320;

let height = 240;

let face;

let day;

function setup() {

canvas = createCanvas(width\*2, height);

let constraints = {

audio: false,

video: {

facingMode: "user"

}

};

capture = createCapture(constraints);

container = select('#container');

takePictureButton = select('#takepicture');

detectFaceButton = select('#detectface');

firstPictureButton = select('#firstpicture');

hasFace = select('#hasFace');

hasUploaded = select('#hasUploaded');

day = select('#day');

face = createImage(width, height);

capture.parent(container);

canvas.parent(container);

capture.size(width, height);

capture.hide();

takePictureButton.mousePressed(takePicture);

detectFaceButton.mousePressed(detectFace);

firstPictureButton.mousePressed(firstPicture);

}

//// Uploading First Picture to the S3 Bucket ///////////

function firstPicture() {

console.log("Uploading First Face");

let intern\_name = select('#intern\_name').value();

console.log(intern\_name);

if (intern\_name == '') {

hasUploaded.html("Please specify the intern's name");

return null;

}

face.loadPixels();

let faceInBase64 = face.canvas.toDataURL("image/png");

faceInBase64 = faceInBase64.replace(/^data:image\/(png|jpg);base64,/, "");

let postData = {

imgstr: faceInBase64,

intern\_name: intern\_name

}

httpPost(FACE\_UPLOAD\_URL, 'json',

postData,

function (result) {

console.log('result')

hasUploaded.html(result);

console.log(result);

;

},

function (error) {

console.log('error');

console.log(error);

}

);

}

/// Check if the uploaded face contains a picture or not ///////////

function detectFace() {

// call the api by sending that picture with it

console.log("Detecting Face.......");

hasFace.html("Detecting Face. Please Wait.......");

face.loadPixels();

let faceInBase64 = face.canvas.toDataURL("image/png");

faceInBase64 = faceInBase64.replace(/^data:image\/(png|jpg);base64,/, "");

let daynumber = day.value();

let postData = {

imgstr: faceInBase64,

test: 1,

day: daynumber

}

httpPost(FACE\_RECOGNITION\_URL, 'json',

postData,

function (result) {

console.log('result')

hasFace.html(result);

console.log(result);

;

},

function (error) {

console.log('error');

console.log(error);

}

);

}

function takePicture() {

console.log('Taking Picture');

face = capture.get(0, 0, width, height);

}

function draw() {

background(255);

image(face, width, 0, width, height);

push();

translate(capture.width,0);

scale(-1.0,1.0);

image(capture.get(), 0, 0, width, height);

pop();

//filter(BLUR, 3);

filter(INVERT);

}

**AWS Lambda:**

import json

import boto3

import random

from boto3.dynamodb.conditions import Key, Attr

########### Setting up the DynamoDB ################

dynamodb = boto3.resource('dynamodb')

table = dynamodb.Table('interns')

email\_table = dynamodb.Table('intern\_emails')

NUM\_OF\_DAYS = 30

PASSWORD\_LENGTH = 5

CHOICES = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '0']

ADMIN\_USERNAME = '\*\*\*'

ADMIN\_PASSWORD = '\*\*\*\*'

############# Helpers.py ###########################

####################################################

def generate\_password():

return ''.join([random.choice(CHOICES) for \_ in range(PASSWORD\_LENGTH)])

def add\_new\_intern(intern\_name, email):

##### use dynamodb and add num\_of\_days # of the items for the intern

with table.batch\_writer() as batch:

for i in range(NUM\_OF\_DAYS):

batch.put\_item(

Item={

'intern\_name': intern\_name,

'day\_number': i + 1,

'password': generate\_password()

}

)

email\_table.put\_item(

Item={

'intern\_name': intern\_name,

'email': email

}

)

def get\_unique\_intern\_names\_from\_dynamo():

response = table.scan(

FilterExpression=Attr('day\_number').eq(1)

)

items = response['Items']

unique\_names = set(item.get('intern\_name') for item in items)

return unique\_names

def get\_attendance\_list(unique\_intern\_names):

### get the attendance lists for every intern from dynamodb table ##

### only returns data where the present value is True

attendance\_list = []

for intern\_name in unique\_intern\_names:

response = table.scan(

FilterExpression=Attr('present').eq(True) & Attr('intern\_name').eq(intern\_name)

)

items = response['Items']

present\_ones = [item.get('day\_number') for item in items]

list\_item = (intern\_name, present\_ones)

attendance\_list.append(list\_item)

return attendance\_list

def get\_password\_list(unique\_intern\_names, day):

password\_list = []

for intern\_name in unique\_intern\_names:

response = table.get\_item(

Key={

'intern\_name': intern\_name,

'day\_number': day

}

)

item = response['Item']

password = item.get('password')

password\_item = (intern\_name, password)

password\_list.append(password\_item)

return password\_list

def get\_task\_list(unique\_intern\_names, day):

task\_list = []

for intern\_name in unique\_intern\_names:

response = table.get\_item(

Key={

'intern\_name': intern\_name,

'day\_number': day

}

)

item = response['Item']

task = {

'yesterday': item.get('yesterday'),

'today': item.get('today')

}

task\_item = (intern\_name, task)

task\_list.append(task\_item)

return task\_list

def get\_password\_for\_intern\_for\_the\_day(username, day):

response = table.get\_item(

Key={

'intern\_name': username,

'day\_number': day

}

)

item = response['Item']

return item.get('password')

def is\_the\_intern\_here(username):

## CHECK IF THE USERNAME IS IN THE DATABASE

# return True or False

response = table.query(

KeyConditionExpression=Key('intern\_name').eq(username)

)

items = response['Items']

if len(items) == 0:

return False

else:

return True

def count\_intern\_attendane\_for\_the\_day(username, day):

# no returns only set the present value to true

table.update\_item(

Key={

'intern\_name': username,

'day\_number': day

},

UpdateExpression='SET present = :val1',

ExpressionAttributeValues={

':val1': True

}

)

def set\_yesterday\_today\_task(username, day, yesterday, today):

# no returns

table.update\_item(

Key={

'intern\_name': username,

'day\_number': day

},

UpdateExpression='SET yesterday = :val1, today = :val2',

ExpressionAttributeValues={

':val1': yesterday,

':val2': today

}

)

def find\_total\_present\_days(username):

## returns a number

response = table.scan(

FilterExpression=Attr('present').eq(True) & Attr('intern\_name').eq(username)

)

items = response['Items']

return len(items)

def get\_intern\_email(intern\_name):

response = email\_table.get\_item(

Key={

'intern\_name': intern\_name

}

)

try:

item = response['Item']

email = item.get('email')

except:

return None

else:

return email

############## The Lambda Handler ############################

def lambda\_handler(event, context):

day = event['day']

try:

day = int(day)

except:

day = ''

add\_mode = event['addMode']

#try:

username = event['username']

password = event['password']

yesterday = event['yesterday']

today = event['today']

email = event['email']

# except:

# username = 'intern'

# password = ''

# yesterday = ''

# today = ''

if add\_mode == "AllSetToGo" and len(username) > 0:

### code to add new intern here

if email == '':

return "There is no email specified"

add\_new\_intern(username, email)

return "Added New Intern"

#############

if username == 'intern':

# interns\_total\_days = [sum(attendance\_list[intern]) for intern in intern\_list]

# k = interns\_total\_days[0]

# p = interns\_total\_days[1]

# n = interns\_total\_days[2]

unique\_intern\_names = get\_unique\_intern\_names\_from\_dynamo()

attendance\_list = get\_attendance\_list(unique\_intern\_names)

return "The attendance dashboard for all interns.{}".format(attendance\_list)

if not isinstance(day, int):

return "The Day is not provided"

### admin check's today's password for all interns

if username == ADMIN\_USERNAME and password == ADMIN\_PASSWORD:

unique\_intern\_names = get\_unique\_intern\_names\_from\_dynamo()

### look up today's password for all interns

#password\_list = [(intern, user\_password\_list[intern][day]) for intern in intern\_list]

password\_list = get\_password\_list(unique\_intern\_names, day)

### Look up the intern's yesterday done and today to-do tasks

#task\_list = [(intern, yesterday\_today\_task\_list[intern][day]) for intern in intern\_list]

task\_list = get\_task\_list(unique\_intern\_names, day)

return "For Day {}, Passwords are {} and Tasks are {}".format(day, password\_list, task\_list)

### the interns take their passwords from the admin on-site in Ashlesha International and check attendance

if is\_the\_intern\_here(username):

password\_for\_the\_day = get\_password\_for\_intern\_for\_the\_day(username, day)

if password\_for\_the\_day == password:

count\_intern\_attendane\_for\_the\_day(username, day) ### Set the Attendance to 1 for the day

#### set the yesterday and today's task for that day

set\_yesterday\_today\_task(username, day, yesterday, today)

return "Attendance counted today on day {} for {}".format(day, username)

present\_days = find\_total\_present\_days(username)

return "Wrong Password. You've been present for a total of {} days".format(present\_days)

return "Nothing From AWS Lambda Here"