

# THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

Faculty of Science



**Fifth-semester mini-project**

*On*

**Fingerprint Attendance System**

*Guided by*

**Mr. Krishnanand Rastogi**

*In fulfilment of the degree*

*Of*

**Bachelors of Computer Applications**

*In*

**Department of computer Applications**

December, 2019

**Submitted by:**

Ebenezer Isaac– 522025

HajiHuseinali Lokhandwala – 522035

Vraj Kotwala – 522033

**The Maharaja Sayajirao University of Baroda  
Faculty of Science  
Department of Computer Applications.**



# MINI-PROJECT

## Cerberus – Attendance Management System

**Date:** 18<sup>th</sup> December 2019

Ebenezer Isaac

Seat no:522025

HajiHuseinali Lokhandwala

Seat no:522035

Vraj Kotwala

Seat no:522033

Invigilator Signature\_\_\_\_\_

## Contents

|        |   |                |
|--------|---|----------------|
| i.     | Cover Page                              | 00             |
| ii.    | College certificate                     | 00             |
| iii.   | Project completion certificate from MSU | 01             |
| iv.    | Acknowledgement                         | 04             |
| v.     | Abstract                                | 05             |
| vi.    | Business problem                        | 06             |
| vii.   | Project charter                         | <b>07 - 08</b> |
|        | 1. Project Definition                   | 07             |
|        | 2. purpose                              | 07             |
|        | 3. scope of project                     | 08             |
|        | 4. team size                            | 08             |
| viii.  | Project Plan                            | 09-11          |
|        | 1. Objective                            | 09             |
|        | 2. Software model agile                 | 09             |
|        | 3. Work division (w.r.t time)           | 09             |
|        | 4. Work division (w.r.t team members)   | 10-11          |
| ix.    | Hardware and Software requirements      | 11             |
| x.     | Tools and technologies used             | 12             |
| xi.    | Product backlogs                        | 13-14          |
| xii.   | Entity-Relationship Diagram             | 15             |
| xiii.  | Database-Dictionary (for ERD)           | 16-21          |
| xiv.   | Sprint Backlogs w burn-down chart       | 23-32          |
| xv.    | Interface Design                        | <b>33-44</b>   |
|        | 1. Screenshots and Explanations         |                |
| xvi.   | Testing                                 | 45-48          |
|        | 1. Test plan                            |                |
|        | 2. Test methods                         |                |
|        | 3. Test cases                           |                |
|        | 4. Bug fixing/ test reports             |                |
| xvii.  | Security features                       | 49             |
| xviii. | Coding standard followed                | 50             |
| xix.   | Limitations and future enhancements     | 51             |
| xx.    | Experience and learning                 | 52             |
| xxi.   | References and Bibliography             | 53-55          |

## Acknowledgment

Success cannot be achieved single-handedly. For this reason, we would like to express our deepest gratitude to all those involved in making this project a success. This project would not have been successfully completed without the help of our mentors.

Firstly, we thank **Prof. R.S. Srivastava (H.O.D.)** who gave us this opportunity to work for the project which will add to our experience and for giving us a chance to prove our knowledge skill and abilities.

We would also like to thank **Mr. Krishnanand Rastogi** who provided us guidelines throughout the course of the project.

Not missing to thank our own **C.A. Department**, for giving us this opportunity to gain practical experience and enable us to understand the demand of the experience and professional environment. We have gained valuable experience which has given us confidence and has enhanced our professional skill for our competent career.

We would also like to thank all our teachers and family members and friends for helping, bearing, encouraging and boosting our morale, throughout our studies, especially during the project period. Last but not the least we would like to express our sincere gratitude to those who helped us in completing our project directly or indirectly.

Special Thanks to **Mr. Rajesh Kotwala** for his help and support for making the box-design possible.

## **Abstract**

This project uses a biometric concept to simplify the attendance system in educational institutes. It uses the most reliable way of distinctively identifying students through fingerprint reading. Such type of application is very useful in school as well as in college for daily attendance. Through this application we can keep a systematic track of student's attendance. This project enables the easy way of maintaining class attendance with fewer efforts.

Also, it has been implemented for the teaching staff.

### **Description:**

- The project requires a fingerprint reader for finger detection.
- Every student can login to the system through finger detection.
- The fingerprint of the student is compared with the one stored in database and if it matches then attendance is marked for that particular student.
- The system also generates a brief report of attendance from the database according to subject-wise or date-wise as required.
- A defaulter list can be generated through system.
- Admin has the option to take a print of the reports and defaulter list thus generated.

### **Advantages:**

- Students will be more regular in attending their classes since now no password or no attendance sheet signature is required, so no friend or any other student can make an attendance on behalf of others as fingerprints are unique for every student.
- Teachers do not need to waste their time approximately 15min of 1 hour for taking attendance of students.
- No need to maintain attendance sheet as the attendance are electronically stored in database.
- The system helps the faculty to easily find out defaulters.
- User may easily get attendance history of a particular student.
- It saves time, cost, efforts and institute resources.

### **Disadvantages:**

- The only disadvantage is that every class requires a fingerprint reader to access the system.

## **Business Problem and Solution**

### *What is the problem?*

Many institutes still follow traditional attendance marking systems which are known for their inconsistencies, discrepancies and unnecessary work force.

### *What is the solution/concept?*

This Fully Automated Attendance System promises to remove all these problems currently faced and install a user-friendly system in its place.

### *What is unique/innovative about the Idea/Concept?*

The Fully Automated system claims a flexible environment for the administrators to change parameters for the description generating system and personalized report for users involved with it.

### *What is the target market or who do you think will benefit from the solution?*

Educational Institutes, Medium Scale to Large Scale Industries/ Offices will be the most benefited from this system, while this system can be implemented in other environments too where the recording attendance of staff/students/employees is required.

### *Detailed Description of the Idea/Concept*

Contrary to the traditional attendance marking system which requires manual labour and risks depreciating accuracy, this system proposes a failsafe attendance marking system. Integrated with an intelligent report generating features, this system promises an easy and user-friendly environment for the administrators.

### *How will you develop and execute the solution?*

The desired objective is achieved with the use of microprocessor(RaspberryPi) to capture fingerprint data and send them to a web server, securely and spontaneously. The web server stores the data into the database for further report generating purposes. A robust and user-friendly graphic user interface is provided for the stakeholders to interact with the system easily in the form of a website. The website promises to generate reports according to the requirements of the administrators on demand.

### *Will your idea/solution impact the population and economy?*

Installing this system in possible locations can improve work efficiency of the employees and removes discrepancies arising due to manual error. It saves manual labour which helps the employee to perform on other productive work.

## Project Charter

### **Project Definition:**

Our project focuses on managing a distributed system of devices that are capable of scanning biometrics for the use of attendance for a large scope of stakeholders automatically on the basis of a timetable. The student will be marked present through biometric (fingerprint) **module** and the attendance data can be accessed on our **website** by the users.

### **Purpose:**

Biometric attendance system has numerous benefits. With the System, maintenance of daily attendance on Excel sheets could be avoided, man labour can be reduced, monthly attendance sheet for employees in Excel can be assembled with a single click, etc.

ERP applications like Cerberus is coded with sophisticated algorithms for attendance management to take-in and track attendance of every individual entering and leaving an organization.

- Biometric systems provide more accurate identification
- Biometric responds rapidly, commonly identifying a candidate in only seconds
- Biometric System is less tedious, easy to use, hard to adulterate.
- Organizations can midway and precisely screen every candidate's attendance to counteract proxy attendance and errors.

**Scope:**

- Scope: Enroll fingerprints
- Authentication
- Manage Faculty/ Admin (Fingerprint data)
- Manage Student (Fingerprint data)

**Background Activities (Sync):**

- Attendance to database
- Fetch templates from database
- Fetch timetable from database.
- Fetch PRN

**Scope of Web-App (made with Java-Servlets):****For Faculty:**

- Login
- Timetable Management
- Subjects Management
- Student Management
- Attendance Management (Manual)
- Admin Management
- Student Progression
- OTP for devices
- Profile Management

**For Students:**

- Login
- View Attendance
- Select Electives

**Team Size: (3 Students).**

Ebenezer Isaac **522025**

Haji Huseinali Lokhandwala **522035**

Vraj Kotwala **522033**



## Project Plan

### **Objective:**

To develop a fully automated Attendance Management System for a University Dept.

### **Software Model:** Agile.

To iterate the development and testing throughout the software development lifecycle of the project.

### **Methodology:** XP

Extreme programming (XP) is a software development practice which is intended to improve software quality and responsiveness to changing customer requirements.

### **Work Division** (w.r.t team members):

- ~~Arduino and circuit design: Ebenezer and Vraj~~
- RaspberryPi: Ebenezer
- Box design: Vraj & Ebenezer.
- Backend: Ebenezer & Vraj
- Front end & GUI: Huseinali
- Database Structure Design & SQL: Vraj
- Documentation: Vraj
- Testing: Vraj, Ebenezer, Huseinali

### **Work Division** (w.r.t time):

#### May:

- Testing components/ Modules.
- FPS, LCD Libraries
- Custom functions for RaspberryPi (waitForFinger(), lcd.print(), lcd.clrscr())
- Database connection with RaspberryPi
- Launcher functions for RaspberryPi

#### June:

- Sync fingerprints
- Switching to GitHub for version control.
- Commercializing the project
- Circuit Diagram for Arduino/ RaspberryPi
- Blob/Clob for finger data
- Database Schema

July:

- View Attendance
- View Timetable
- Edit Timetable
- Manage password
- Password encryption (SHA2)
- Edit timings
- Add slot
- Remove slot
- View Student Details
- Student Subject Selection
- Add Student
- Delete Student

August:

- Manage Subjects (Add, View, Delete)
- Faculty Management (Add, View, Delete)
- Final Database Normalization
- Adding logs
- Dividing storage space into classes
- Simultaneous interaction of SPI Devices.
- Ethernet Sleep mode.
- Add light show for standby

September

- Download data to excel (Backup)
- Upload student data from excel file.
- Profile Page
- Activity log
- Session Management
- ReCAPTCHA

October:

- Mailer
- OTP
- Email verification
- Handle SQL Exceptions
- Validations in forms (AJAX)
- Animations using CSS and JS
- Particle.js
- Theme and logo

November:

- Box Design
- Student Progression
- JSON format data
- Testing
- About Us page

December:

- Copy timetable from previous week
- Fix scrollbar
- First login asks for details
- Manual Attendance
- Email and network connectivity check
- Documentation
- Presentation

**Hardware Requirements:**

- RaspberryPi Microcontroller
- FPS Module
- Ethernet Module
- Buzzer
- Real Time Clock (RTC)
- Screen
- Keypad
- HDD space on server (for database): 12 MB
- HDD space on RaspberryPi: 8 GB memory card

**Software Requirements:**

- OS: Windows 7 or above on server
- OS: RaspbianOS on RaspberryPi.
- PhpMyAdmin (4.6 or above)

### **Tools and Technologies used:**

#### **On Device (RaspberryPi):**

- OS: Raspbian
- Lang: Python
- Tool: Notepad++

#### **On Web-App:**

##### **frontend:**

- AJAX
- HTML5
- JS
- CSS

##### **backend:**

- Java-Servlet
- JSP.

#### **Database:**

MySQL with PhpMyAdmin

#### **Server:**

Glassfish Server 4.1.1

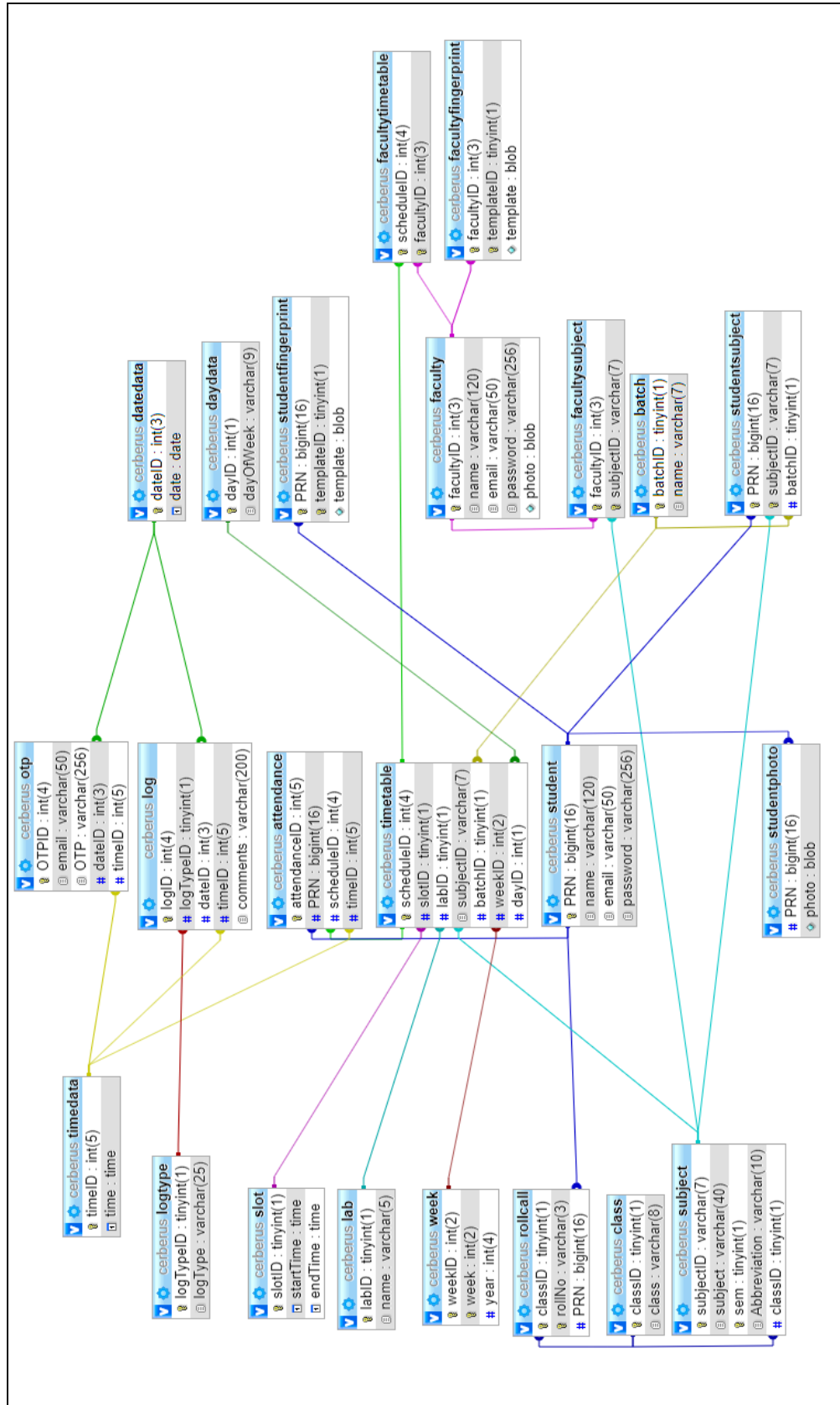
## **Product Backlogs:**

The product backlog is the single authoritative source for things that a team works on.

- Testing components/ Modules.
- FPS, LCD Libraries
- **Custom functions for RaspberryPi (waitForFinger(), lcd.print(), lcd.clrscr())**
- Database connection with RaspberryPi
- Launcher functions for RaspberryPi
  
- Sync fingerprints
- Switching to GitHub for version control.
- **Commercializing the project**
- **Circuit Diagram for Arduino/ RaspberryPi**
- Blob/Clob for finger data
- **Database Schema**
  
- View Attendance
- View Timetable
- Edit Timetable
- Manage password
- Password encryption (SHA2)
- Edit timings
- Add slot
- Remove slot
- View Student Details
- Student Subject Selection
- Add Student
- Delete Student
  
- Manage Subjects (Add, View, Delete)
- Faculty Management (Add, View, Delete)
- **Final Database Normalization**
- Adding logs
- Dividing storage space into classes
- Simultaneous interaction of SPI Devices.
- Ethernet Sleep mode.
- Add light show for standby
  
- Download data to excel (Backup)
- Upload student data from excel file.
- Profile Page
- Activity log
- Session Management
- ReCAPTCHA

- Mailer
- OTP
- Email verification
- Handle SQL Exceptions
- Validations in forms (AJAX)
- Animations using CSS and JS
- Particle.js
- Theme and logo
  
- **Box Design**
- **Student Progression**
- JSON format data
- **Testing**
- About Us page
  
- Copy timetable from previous week
- Fix scrollbar
- First login asks for details
- **Manual Attendance**
- Email and network connectivity check
- **Documentation**
- **Presentation**

## ER-Diagram



### Database-Dictionary:

#### attendance

| Column                          | Type       | Null | Default | Comments |
|---------------------------------|------------|------|---------|----------|
| attendanceID ( <i>Primary</i> ) | int(5)     | No   |         |          |
| PRN                             | bigint(16) | No   |         |          |
| scheduleID                      | int(4)     | No   |         |          |
| timeID                          | int(5)     | No   |         |          |

#### batch

| Column                     | Type       | Null | Default | Comments |
|----------------------------|------------|------|---------|----------|
| batchID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| name                       | varchar(7) | No   |         |          |

#### class

| Column                     | Type       | Null | Default | Comments |
|----------------------------|------------|------|---------|----------|
| classID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| class                      | varchar(8) | No   |         |          |

#### datedata

| Column                    | Type   | Null | Default | Comments |
|---------------------------|--------|------|---------|----------|
| dateID ( <i>Primary</i> ) | int(3) | No   |         |          |
| date                      | date   | No   |         |          |



### daydata

| Column                   | Type       | Null | Default | Comments |
|--------------------------|------------|------|---------|----------|
| dayID ( <i>Primary</i> ) | int(1)     | No   |         |          |
| dayOfWeek                | varchar(9) | No   |         |          |

### faculty

| Column                       | Type         | Null | Default | Comments |
|------------------------------|--------------|------|---------|----------|
| facultyID ( <i>Primary</i> ) | int(3)       | No   |         |          |
| name                         | varchar(120) | No   |         |          |
| email                        | varchar(50)  | No   |         |          |
| password                     | varchar(256) | No   |         |          |
| photo                        | blob         | No   |         |          |

### facultyfingerprint

| Column                        | Type       | Null | Default | Comments |
|-------------------------------|------------|------|---------|----------|
| facultyID ( <i>Primary</i> )  | int(3)     | No   |         |          |
| templateID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| template                      | blob       | Yes  | NULL    |          |

### facultysubject

| Column                       | Type       | Null | Default | Comments |
|------------------------------|------------|------|---------|----------|
| facultyID ( <i>Primary</i> ) | int(3)     | No   |         |          |
| subjectID ( <i>Primary</i> ) | varchar(7) | No   |         |          |

### facultytimetable

| Column                        | Type   | Null | Default | Comments |
|-------------------------------|--------|------|---------|----------|
| scheduleID ( <i>Primary</i> ) | int(4) | No   |         |          |
| facultyID ( <i>Primary</i> )  | int(3) | No   |         |          |

### lab

| Column                   | Type       | Null | Default | Comments |
|--------------------------|------------|------|---------|----------|
| labID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| name                     | varchar(5) | No   |         |          |

### log

| Column                   | Type         | Null | Default | Comments |
|--------------------------|--------------|------|---------|----------|
| logID ( <i>Primary</i> ) | int(4)       | No   |         |          |
| logTypeID                | tinyint(1)   | No   |         |          |
| dateID                   | int(3)       | No   |         |          |
| timeID                   | int(5)       | No   |         |          |
| comments                 | varchar(200) | No   |         |          |

### logtype

| Column                       | Type        | Null | Default | Comments |
|------------------------------|-------------|------|---------|----------|
| logTypeID ( <i>Primary</i> ) | tinyint(1)  | No   |         |          |
| logType                      | varchar(25) | No   |         |          |

## otp

| Column                   | Type         | Null | Default | Comments |
|--------------------------|--------------|------|---------|----------|
| OTPID ( <i>Primary</i> ) | int(4)       | No   |         |          |
| email                    | varchar(50)  | No   |         |          |
| OTP                      | varchar(256) | No   |         |          |
| dateID                   | int(3)       | No   |         |          |
| timeID                   | int(5)       | No   |         |          |

## rollcall

| Column                     | Type       | Null | Default | Comments |
|----------------------------|------------|------|---------|----------|
| classID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| rollNo ( <i>Primary</i> )  | varchar(3) | No   |         |          |
| PRN                        | bigint(16) | No   |         |          |

## slot

| Column                    | Type       | Null | Default | Comments |
|---------------------------|------------|------|---------|----------|
| slotID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| startTime                 | time       | No   |         |          |
| endTime                   | time       | No   |         |          |

### student

| Column                 | Type         | Null | Default | Comments |
|------------------------|--------------|------|---------|----------|
| PRN ( <i>Primary</i> ) | bigint(16)   | No   |         |          |
| name                   | varchar(120) | No   |         |          |
| email                  | varchar(50)  | No   |         |          |
| password               | varchar(256) | No   |         |          |

### studentfingerprint

| Column                        | Type       | Null | Default | Comments |
|-------------------------------|------------|------|---------|----------|
| PRN ( <i>Primary</i> )        | bigint(16) | No   |         |          |
| templateID ( <i>Primary</i> ) | tinyint(1) | No   |         |          |
| template                      | blob       | Yes  | NULL    |          |

### studentphoto

| Column | Type        | Null | Default | Comments |
|--------|-------------|------|---------|----------|
| PRN    | varchar(16) | No   |         |          |
| photo  | blob        | Yes  | NULL    |          |

### Studentsubject

| Column                       | Type       | Null | Default | Comments |
|------------------------------|------------|------|---------|----------|
| PRN ( <i>Primary</i> )       | bigint(16) | No   |         |          |
| subjectID ( <i>Primary</i> ) | varchar(7) | No   |         |          |
| batchID                      | tinyint(1) | No   |         |          |

### subject

| Column                     | Type        | Null | Default | Comments |
|----------------------------|-------------|------|---------|----------|
| subjectID <i>(Primary)</i> | varchar(7)  | No   |         |          |
| subject                    | varchar(40) | No   |         |          |
| sem <i>(Primary)</i>       | tinyint(1)  | No   |         |          |
| Abbreviation               | varchar(10) | No   |         |          |
| classID                    | tinyint(1)  | No   |         |          |

### timedata

| Column                  | Type   | Null | Default | Comments |
|-------------------------|--------|------|---------|----------|
| timeID <i>(Primary)</i> | int(5) | No   |         |          |
| time                    | time   | No   |         |          |

### timetable

| Column                      | Type       | Null | Default | Comments |
|-----------------------------|------------|------|---------|----------|
| scheduleID <i>(Primary)</i> | int(4)     | No   |         |          |
| slotID                      | tinyint(1) | No   |         |          |
| labID                       | tinyint(1) | No   |         |          |
| subjectID                   | varchar(7) | No   |         |          |
| batchID                     | tinyint(1) | No   |         |          |
| weekID                      | int(2)     | No   |         |          |
| dayID                       | int(1)     | No   |         |          |

## week

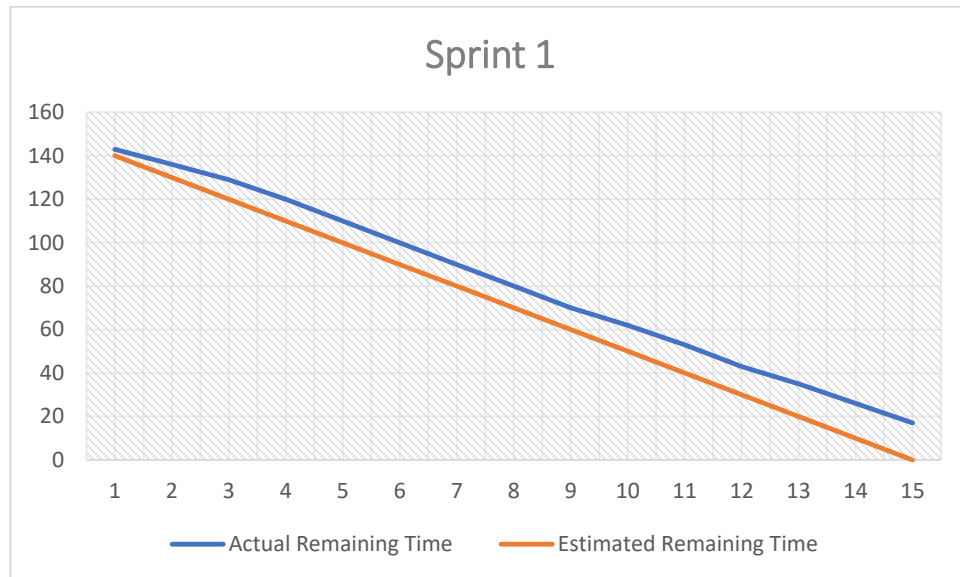
| Column                    | Type   | Null | Default | Comments |
|---------------------------|--------|------|---------|----------|
| weekID ( <i>Primary</i> ) | int(2) | No   |         |          |
| week                      | int(2) | No   |         |          |
| year                      | int(4) | No   |         |          |

## **Sprint Backlogs with Burn-down Charts:**

Sprint1 (July1 to July15)

Stories:

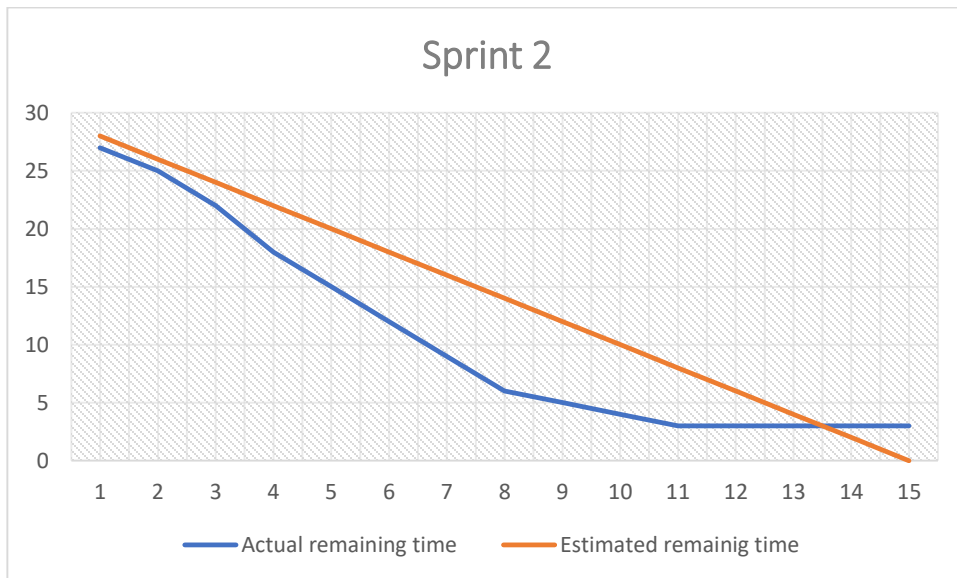
|                             |
|-----------------------------|
| Testing hardware components |
| FPS,LCD Libraries           |



## Sprint2 (July16 to July30)

Stories:

|                                    |
|------------------------------------|
| Custom functions for RaspberryPi   |
| Database connect with RaspberryPi  |
| Launcher functions for RaspberryPi |

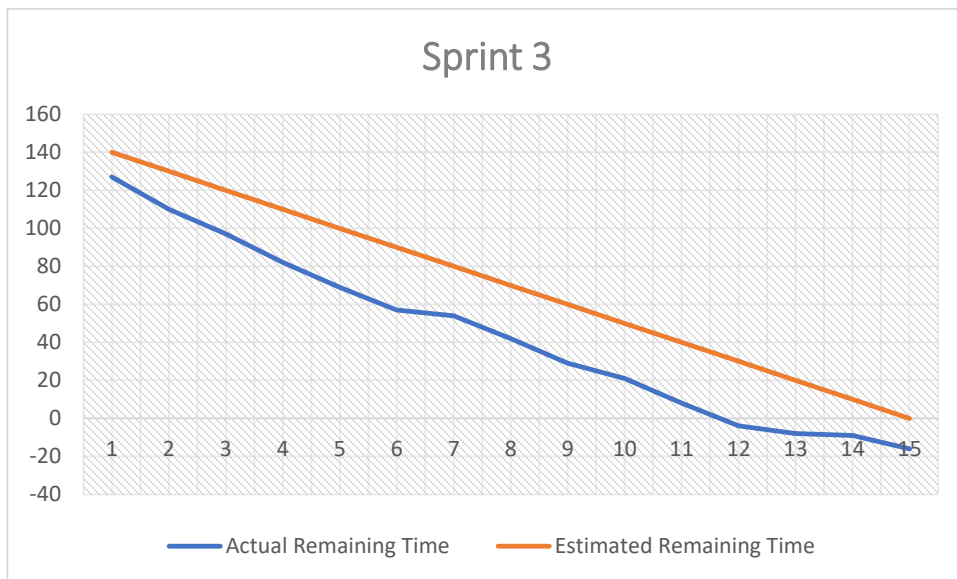




### Sprint3 (Aug1 to Aug15)

Stories:

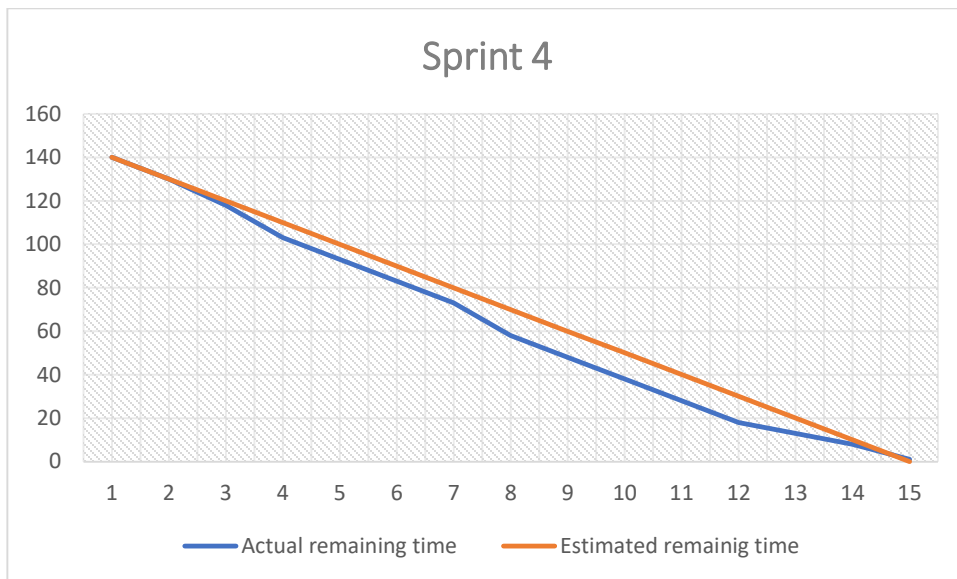
|   |
|---|
| Sync fingerprints                           |
| Switching to GitHub for version control.    |
| Commercializing the project                 |
| Circuit Diagram for Arduino/<br>RaspberryPi |
| Blob/Clob for finger data                   |
| Database Schema                             |



### Sprint4 (Aug16 to Aug30)

Stories:

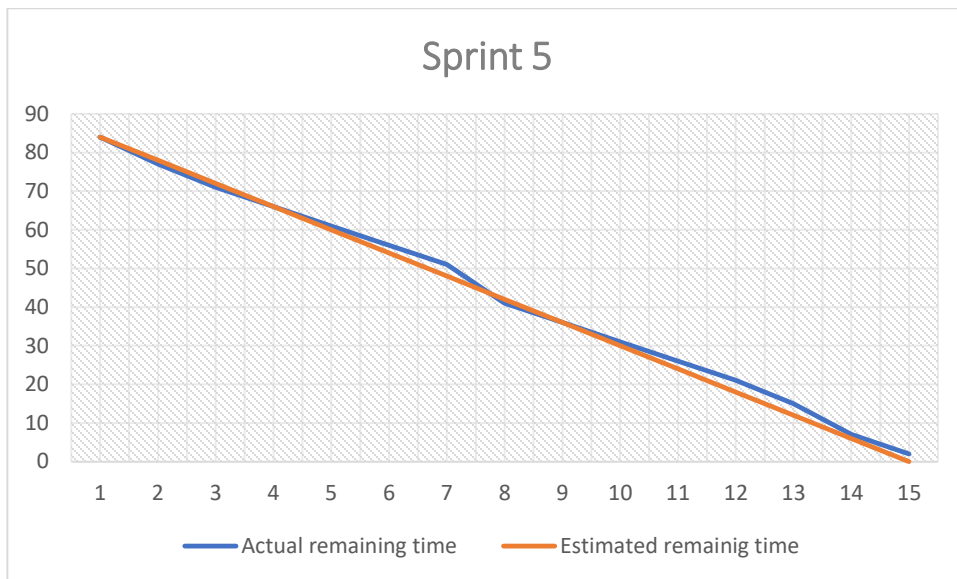
|                            |
|----------------------------|
| View Attendance            |
| View Timetable             |
| Edit Timetable             |
| Manage password            |
| Password encryption (SHA2) |
| Edit timings               |



### Sprint5 (Sep1 to Sep15)

Stories:

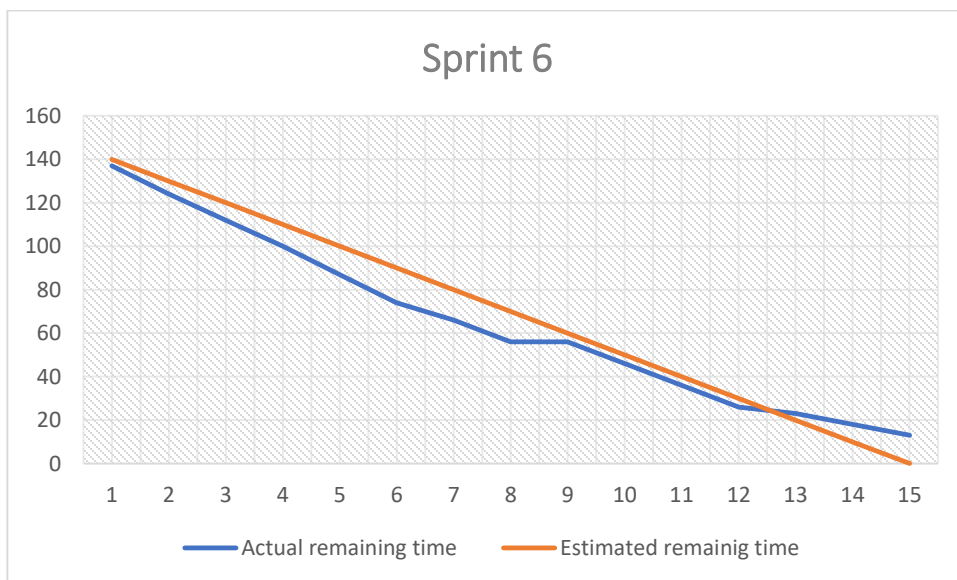
|                           |
|---------------------------|
| Add slot                  |
| Remove slot               |
| View Student Details      |
| Student Subject Selection |
| Add Student               |
| Delete Student            |



### Sprint6 (Sep16 to Sep30)

Stories:

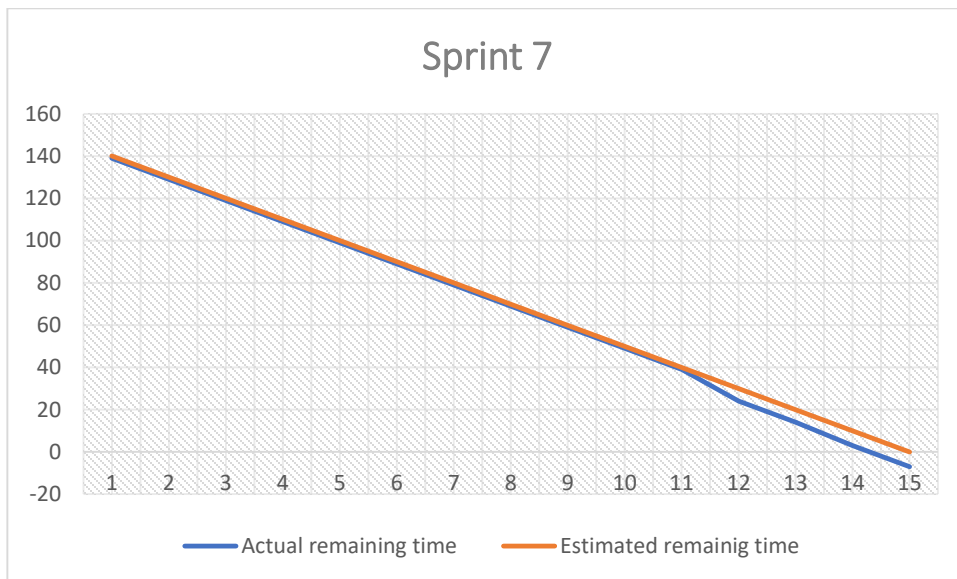
|  |
|--|
| Manage Subjects (Add, View, Delete)      |
| Faculty Management (Add, View, Delete)   |
| Final Database Normalization             |
| Dividing storage space into classes      |
| Simultaneous interaction of SPI Devices. |
| Ethernet Sleep mode.                     |
| Add light show for standby               |



### Sprint7 (Oct1 to Oct15)

Stories:

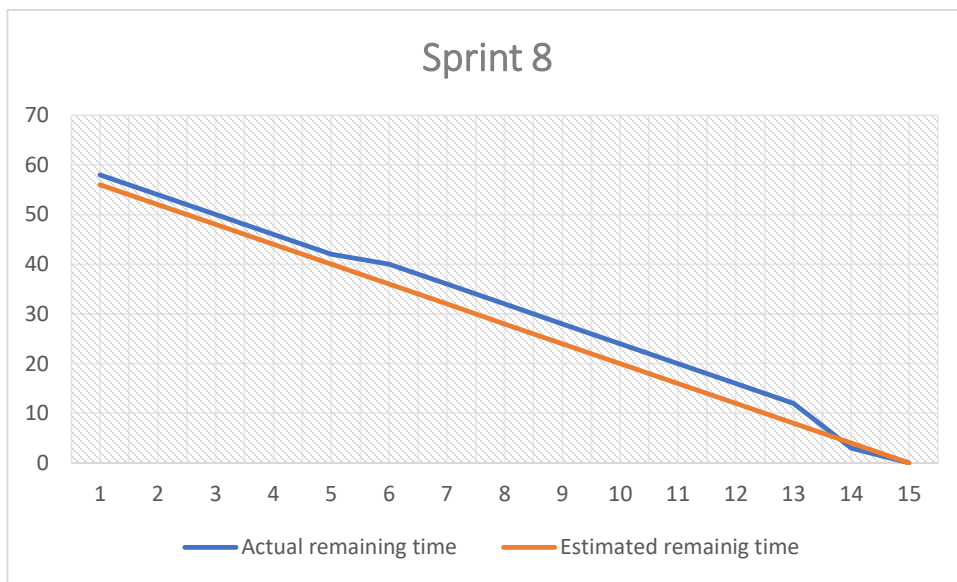
|                                      |
|--------------------------------------|
| Download data to excel (Backup)      |
| Upload student data from excel file. |
| Profile Page                         |
| Activity log                         |
| Session Management                   |
| ReCAPTCHA                            |



### Sprint8 (Oct16 to Oct30)

Stories:

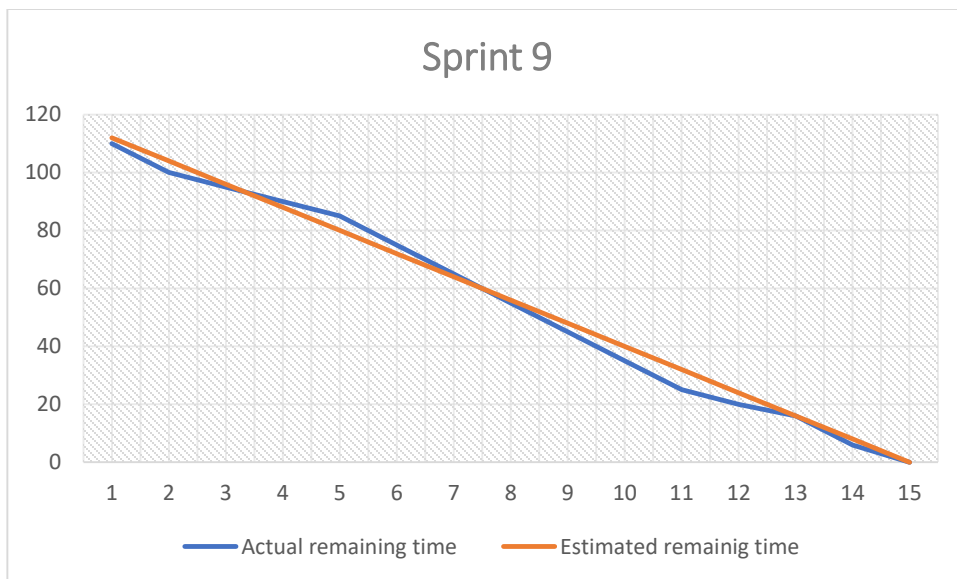
|                             |
|-----------------------------|
| Mailer                      |
| OTP                         |
| Handle SQL Exceptions       |
| Validations in forms (AJAX) |
| Animations using CSS and JS |
| Particle.js                 |
| Theme and logo              |



### Sprint9 (Nov1 to Nov15)

Stories:

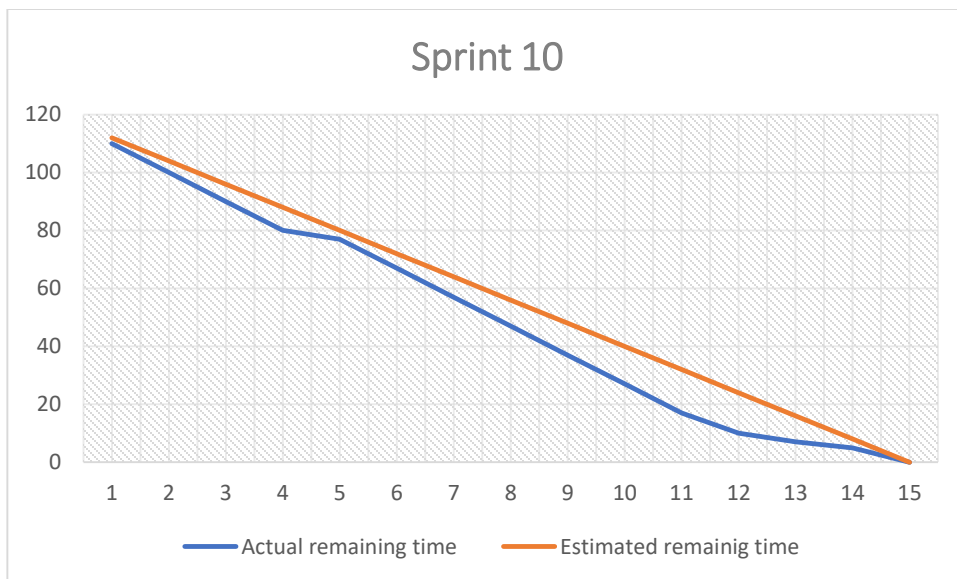
|                      |
|----------------------|
| Box Design           |
| JSON format data     |
| Testing              |
| Ajax on website load |
| About Us page        |



### Sprint10 (Nov16 to Nov30)

Stories:

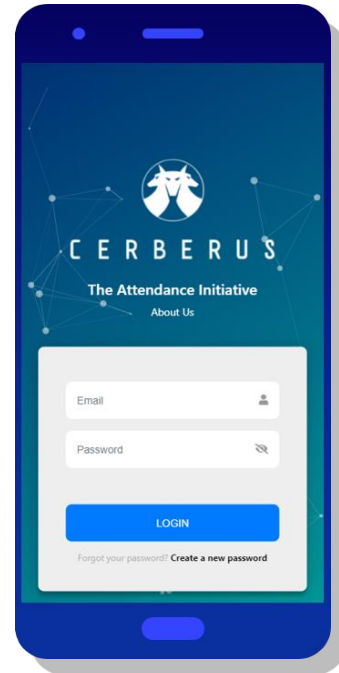
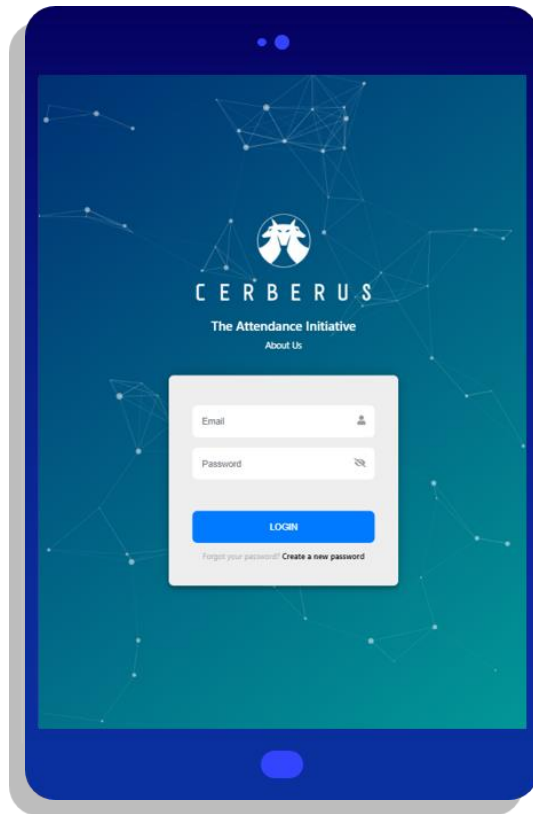
|                                      |
|--------------------------------------|
| Copy timetable from previous week    |
| Fix scrollbar                        |
| First login asks for details         |
| Manual Attendance                    |
| Email and network connectivity check |



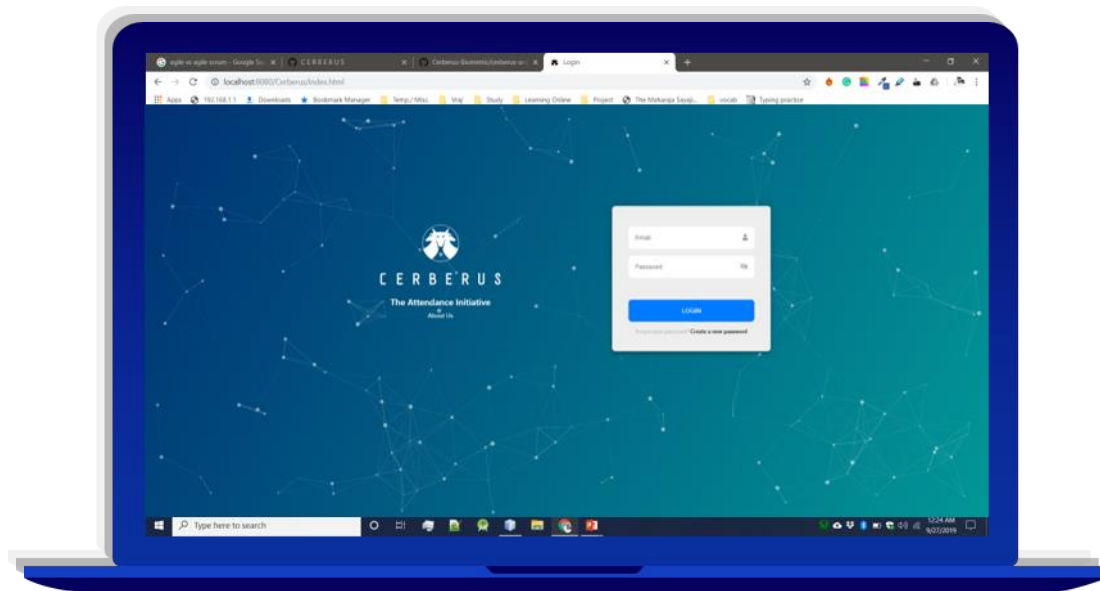


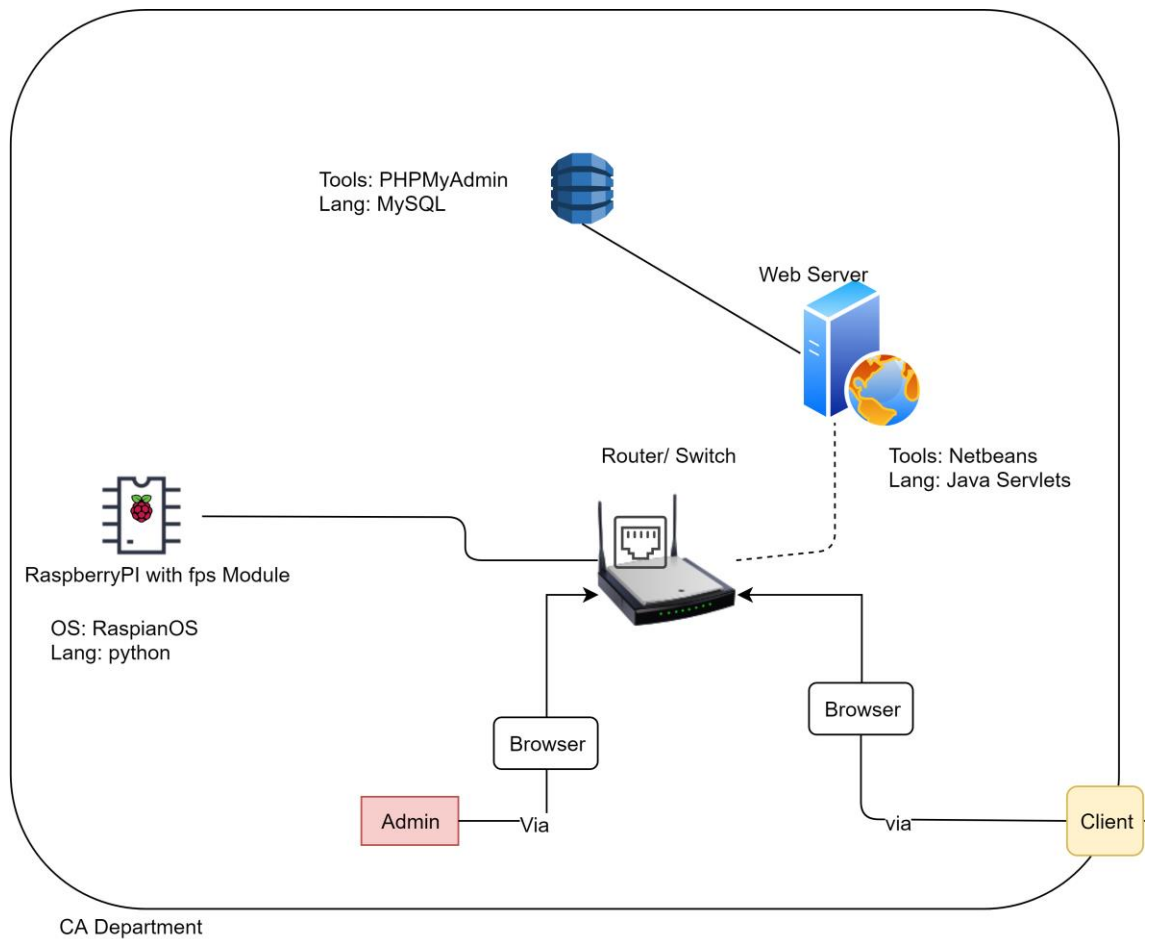
## Interface Design (Screenshots and Explanations)

Perks of using Bootstrap:



Tablet-View  
Mobile-View.  
Web-View





### System Architecture:

- The attendance can be marked by the help of RaspberryPi with FPS module.
- The finger templates are then stored in the database (which resides on web server) with the help of router.
- The webserver also helps retrieve the data when accessed by browser.
- The attendance can be manipulated by Admin/Teachers.
- The attendance can be viewed by Students through portal.

|            |          | Current Week : 51<br>Displaying Timetable of Week : 51 |                       |                         |                        |                      |                        |
|------------|----------|--|-----------------------|-------------------------|------------------------|----------------------|------------------------|
|            |          | LAB 3<br>2019-12-09 to 2019-12-16                      |                       |                         |                        |                      |                        |
| Start Time | End Time | Monday<br>2019-12-10                                   | Tuesday<br>2019-12-11 | Wednesday<br>2019-12-12 | Thursday<br>2019-12-13 | Friday<br>2019-12-14 | Saturday<br>2019-12-15 |
| 08 : 00    | 10 : 00  | No Lab   | No Lab                | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| 10 : 30    | 12 : 00  | No Lab   | No Lab                | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| 13 : 00    | 15 : 00  | No Lab   | No Lab                | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| 15 : 00    | 17 : 00  | No Lab   | No Lab                | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| 17 : 30    | 19 : 30  | No Lab   | No Lab                | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| Start Time | End Time | Monday<br>2019-12-10                                   | Tuesday<br>2019-12-11 | Wednesday<br>2019-12-12 | Thursday<br>2019-12-13 | Friday<br>2019-12-14 | Saturday<br>2019-12-15 |

### Edit Timetable

- Teacher/Admin can edit the week's timetable and store it into database.
- Smooth GUI for easy understanding and manipulation of data is applied.

Student

Subject

Faculty

Student Progression

BCA TY

| Subject Code | Subject Abbr | Batch 1 | Batch 2 | Batch 3 |
|--------------|--------------|---------|---------|---------|
| BCA1501      | XML          | 0       | 0       | 0       |
| BCA1530      | WAD          | 0       | 0       | 0       |
| BCA1538      | AI           | 2       | 0       | 0       |
| BCA1539      | MC           | 0       | 0       | 0       |
| Subject Code | Subject Abbr | Batch 1 | Batch 2 | Batch 3 |

Select Subject :

Select Subject

Edit Attendance

\*Or to Conduct New Lab Session

Student Average Attendance

Show10entries

Search:

| Roll | T1 | Name         | T1 | XML   | T1 | WAD   | T1 | AI    | T1 | MC | T1 |
|------|----|--------------|----|-------|----|-------|----|-------|----|----|----|
| 1    |    | Ebenezer     |    | 0.00% |    | 0.00% |    | 0.00% |    | NA |    |
| 3    |    | Vraj Kotwala |    | NA    |    | NA    |    | 0.00% |    | NA |    |
| Roll |    | Name         |    | XML   |    | WAD   |    | AI    |    | MC |    |

Showing 1 to 2 of 2 entries

Previous

1

Next

### View Attendance

- Total subjects for particular class and their respective lab sessions can be viewed in this page of the portal.
- Manual Attendance can also be done.
- All students' data with respect to class and their selected subject can be viewed.
- Students' percent wise attendance in a particular subject can be viewed too.

Current Week : 51  
Displaying Timetable of Week : 51  
2019-12-09 to 2019-12-16  
LAB 1

| Start Time | End Time | Monday<br>2019-12-10 | Tuesday<br>2019-12-11 | Wednesday<br>2019-12-12 | Thursday<br>2019-12-13 | Friday<br>2019-12-14 | Saturday<br>2019-12-15 |
|------------|----------|----------------------|-----------------------|-------------------------|------------------------|----------------------|------------------------|
| 08:00      | 10:00    | AI<br>Batch A        | No Lab                | No Lab                  | No Lab                 | PCDB<br>Batch A      | DE<br>Batch C          |
| 10:30      | 12:00    | JAVA<br>Batch A      | JAVA<br>Batch C       | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| 13:00      | 15:00    | MC<br>Batch A        | HTML-2<br>Batch C     | No Lab                  | No Lab                 | No Lab               | No Lab                 |
| 15:00      | 17:00    | PCDB<br>Batch A      | No Lab                | No Lab                  | No Lab                 | No Lab               | AI<br>Batch A          |
| 17:30      | 19:30    | No Lab               | No Lab                | No Lab                  | No Lab                 | No Lab               | Python<br>Batch A      |

### View Timetable

- Timetable can be viewed for current and previous weeks.

Start Time      End Time

08:00 AM      10:00 AM

10:30 AM      12:00 PM

01:00 PM      03:00 PM

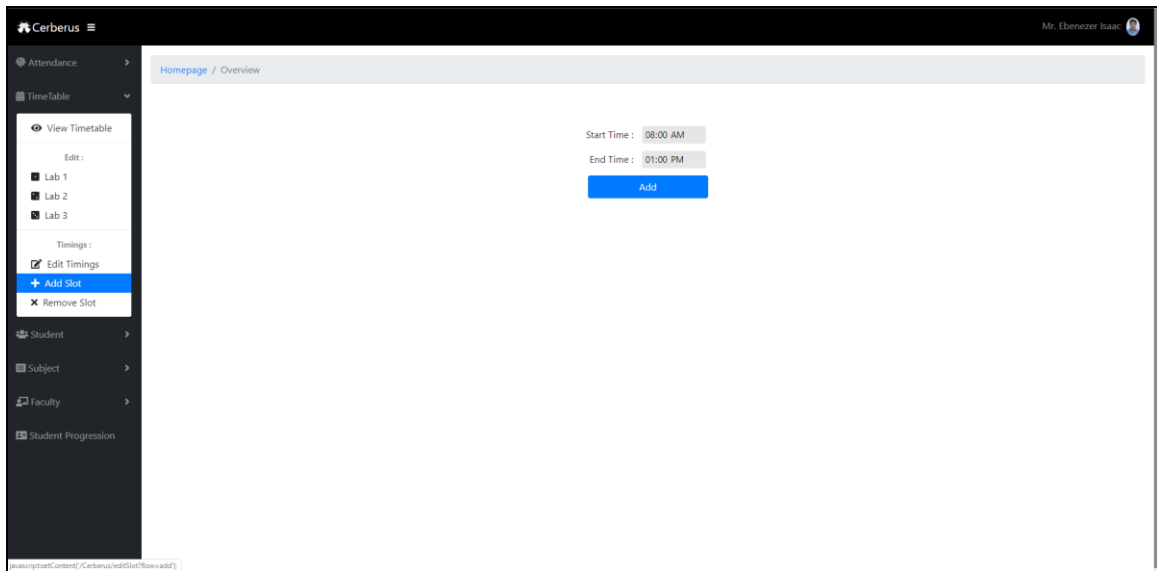
03:00 PM      05:00 PM

05:30 PM      07:30 PM

Submit

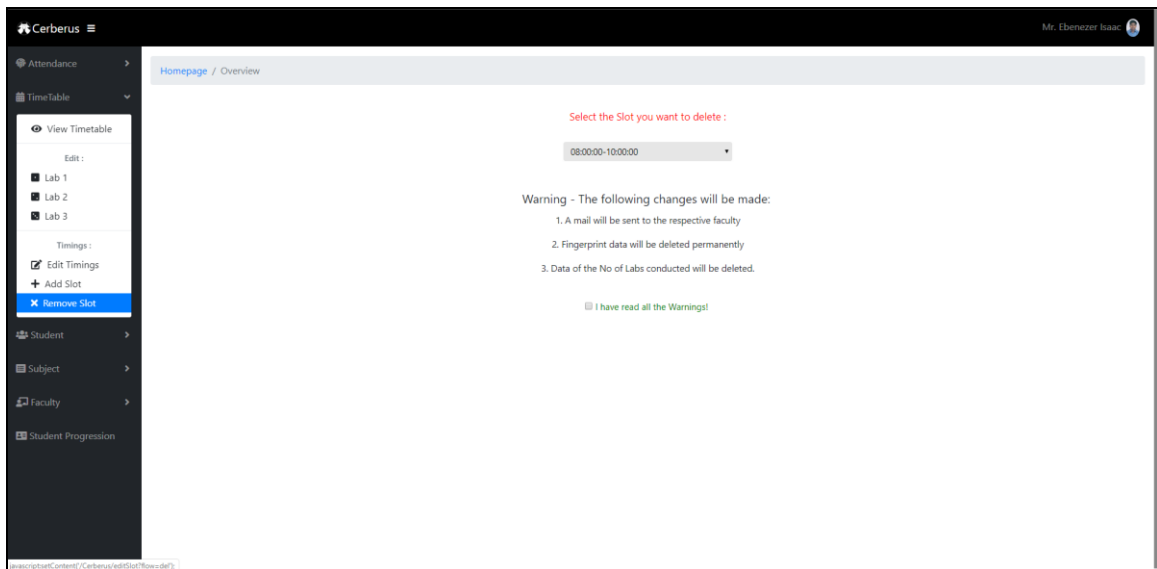
### Edit Timings

- Timings can be added and removed for all the lab sessions of the day.



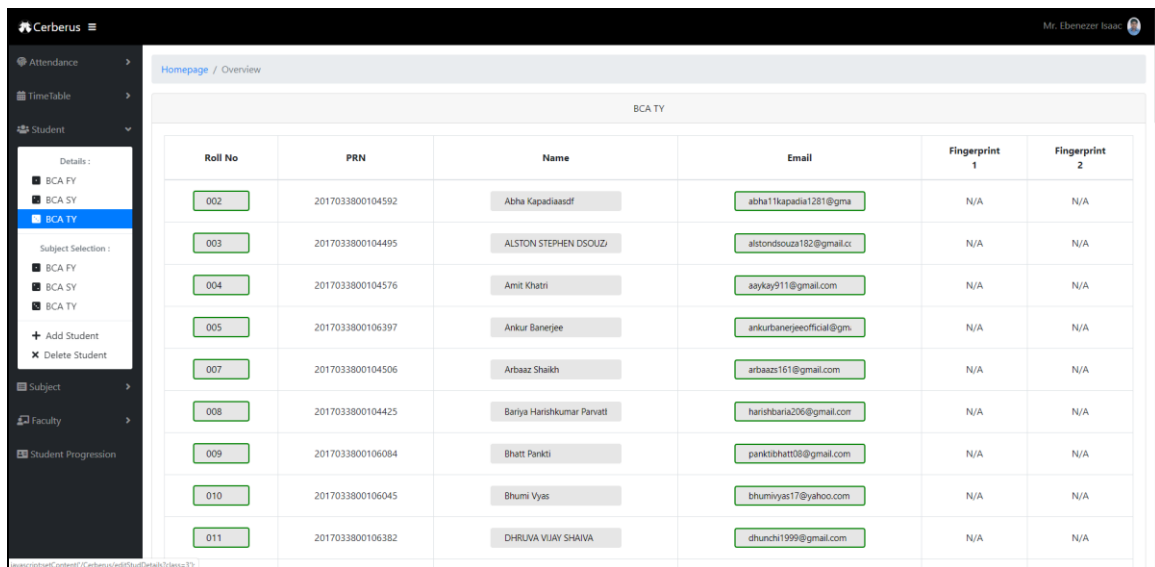
### Add Slot

- Multiple time slots can be added.



### Remove Slot

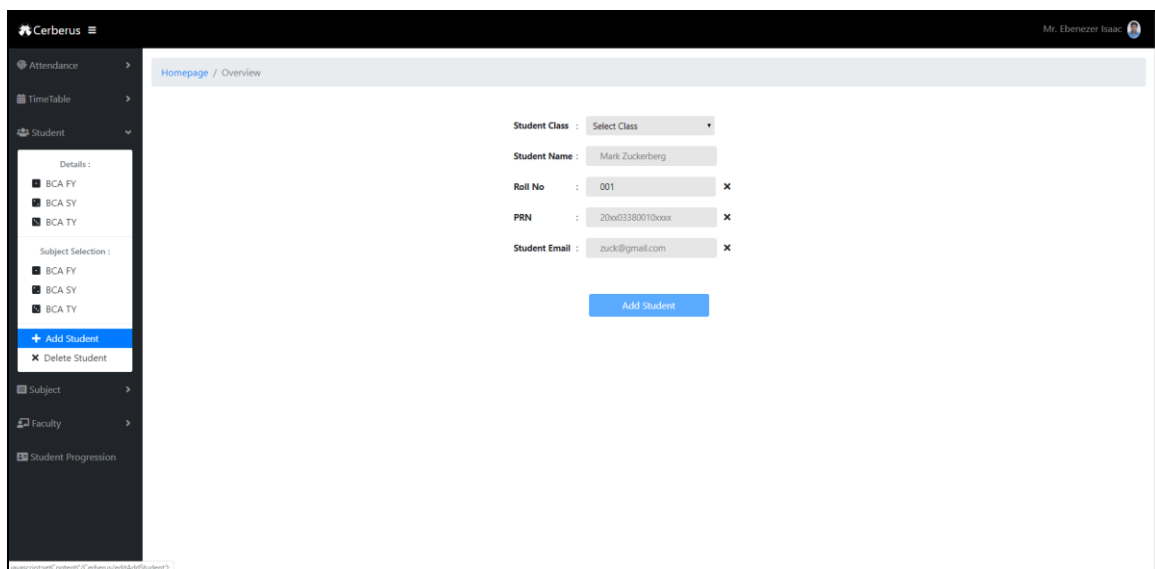
- A time slot can be removed too.



| Roll No | PRN              | Name                       | Email                     | Fingerprint 1 | Fingerprint 2 |
|---------|------------------|----------------------------|---------------------------|---------------|---------------|
| 002     | 2017033800104592 | Abha Kapadiaadrf           | abha11kapadia1281@gma     | N/A           | N/A           |
| 003     | 2017033800104495 | ALSTON STEPHEN DSOUZ       | alstondsouza182@gmail.co  | N/A           | N/A           |
| 004     | 2017033800104576 | Amit Khatri                | aaykay911@gmail.com       | N/A           | N/A           |
| 005     | 2017033800106397 | Ankur Banerjee             | ankurbanerjeeofficial@gm  | N/A           | N/A           |
| 007     | 2017033800104506 | Arbaaz Shaikh              | arbaaz161@gmail.com       | N/A           | N/A           |
| 008     | 2017033800104425 | Baniya Harishkumar Parvati | harishbaria206@gmail.com  | N/A           | N/A           |
| 009     | 2017033800106084 | Bhatt Parikti              | pariktibhatt108@gmail.com | N/A           | N/A           |
| 010     | 2017033800106045 | Bhumi Vyas                 | bhumivyas17@yahoo.com     | N/A           | N/A           |
| 011     | 2017033800106382 | DHRLVA VIJAY SHA'NA        | dhunchi1999@gmail.com     | N/A           | N/A           |

### Student Details

- Student details like roll number, PRN, name, email, and finger template data can be viewed from here.
- Student details are divided into classes.
- Student details can also be edited from here.



Student Class :

Student Name :

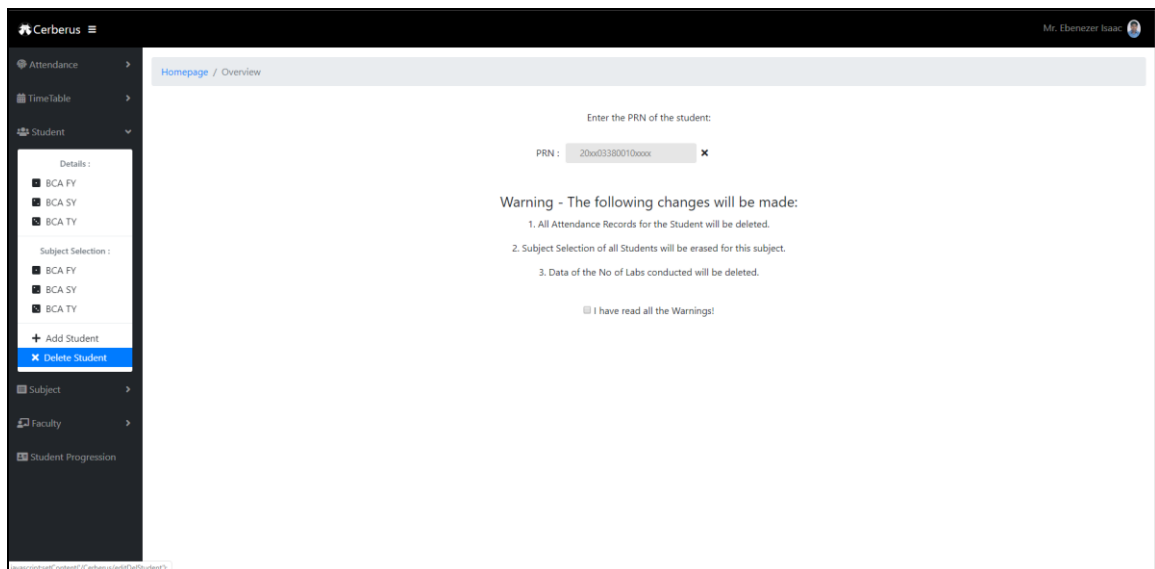
Roll No :

PRN :

Student Email :

### Add Student

- Student can be added from the portal.
- Validations are applied with the help of AJAX.
- A mail is sent to student with OTP for first login when he/she is added from here.



### Delete Student

- Student can be removed from the database as well.
- A mail is sent to the student when he/she is removed.

| Subject Code | Semester | Subject Name                     | Abbreviation | Class  |
|--------------|----------|----------------------------------|--------------|--------|
| BCA1105      | 1        | PC Software and Database         | PCDB         | BCA FY |
| BCA1106      | 1        | Programming                      | Python       | BCA FY |
| BCA1208      | 1        | HTML-I                           | HTML-1       | BCA FY |
| BCA1001      | 2        | Desktop Publishing               | DTP          | BCA FY |
| BCA1205      | 2        | Object Oriented Programming      | OOP          | BCA FY |
| BCA1206      | 2        | Data Structures                  | DS           | BCA FY |
| BCA1207      | 2        | Structured Query Language        | SQL          | BCA FY |
| BCA1301      | 3        | Java                             | JAVA         | BCA SY |
| BCA1304      | 3        | Shell Programming                | SHELL        | BCA SY |
| BCA1305      | 3        | Database Application Programming | DAP          | BCA SY |
| BCA1306      | 3        | Computer Networks-I              | CN-1         | BCA SY |
| BCA1307      | 3        | HTML-II                          | HTML-2       | BCA SY |
| BCA1308      | 3        | Data Exploration                 | DE           | BCA SY |
| BCA1009      | 4        | Web Publishing                   | Web Pub      | BCA SY |

### View Subject

- All the subjects of all the semesters including electives are visible here.
- Lab sessions for particular subject can be seen if the subject linked is further clicked.

### Add Subject

- New subjects if added, then they can be added from here.

### Delete Subject

- Old/Outdated subjects can be removed from the portal from here.



Cerberus

Mr. Ebenezer Isaac

Homepage / Overview

Faculty Name: Mr. First Name

Faculty Email: zuck@gmail.com

Add Faculty

Management:

- Add Faculty
- Delete Faculty

Student Progression

### Add faculty

- New joining faculty can be added from here.
- Mail is received to the respective faculty when he/she is added.

Cerberus

Mr. Ebenezer Isaac

Homepage / Overview

Select the Faculty you want to delete:

Mr.Kutij

Warning - The following changes will be made:

1. A mail will be sent to the respective faculty
2. Fingerprint data will be deleted permanently
3. Data of the No of Labs conducted will be deleted.

☒ I have read all the Warnings!

Submit

### Delete Faculty

- Faculty can be deleted.
- Mail is received and the respective faculty has no longer access to the portal.

**Download Backup**

Please download respective files to backup locally.

**Class Wise Attendance**

DownloadTYData Submit Submit

DownloadSYData DownloadTYData

**Subject Wise Attendance**

DownloadSubjectWiseData Submit

Please Read: All the data will be available in zip format. With Subjects and Batches.

Next

### Student Progression

- When Student passes class and is shifted to higher class, its data is transferred too.
- Backup of attendance is being done before shifting to new sem.
- All the tables in the database are updated
- Change in student roll number after he/she passes class can be done here.
- ClassID for particular student can be manipulated.
- New data for FY Students can be uploaded to database with an excel file.
- New Students are then registered automatically to the portal.
- New students receive the mail with OTP for first time login.
- Students are then asked to enroll their respective fingerprints in RaspberryPi.
- Previous pass out student data is removed from the database with backup kept.
- Student new subject is linked with SY and TY students.

**Cerberus** Mr. Ebenezer Isaac

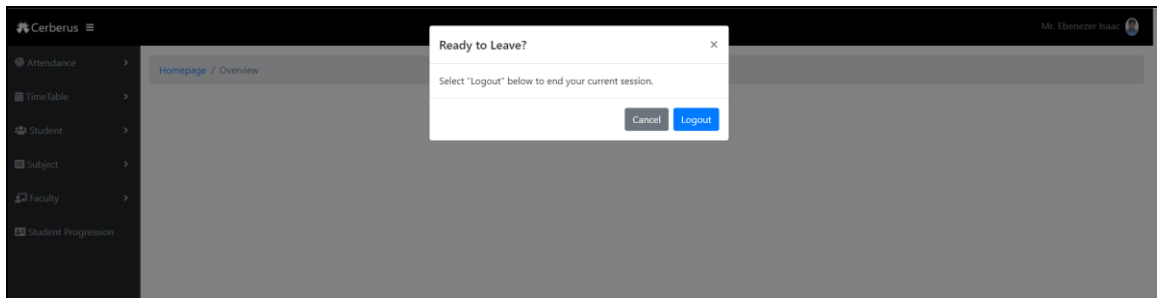
Homepage / Overview

Mr. Ebenezer Isaac

My Preferences

| Subject Code | Abbreviation | Preference |
|--------------|--------------|------------|
| Subject Code | Abbreviation | Preference |

### Student/Admin Profile

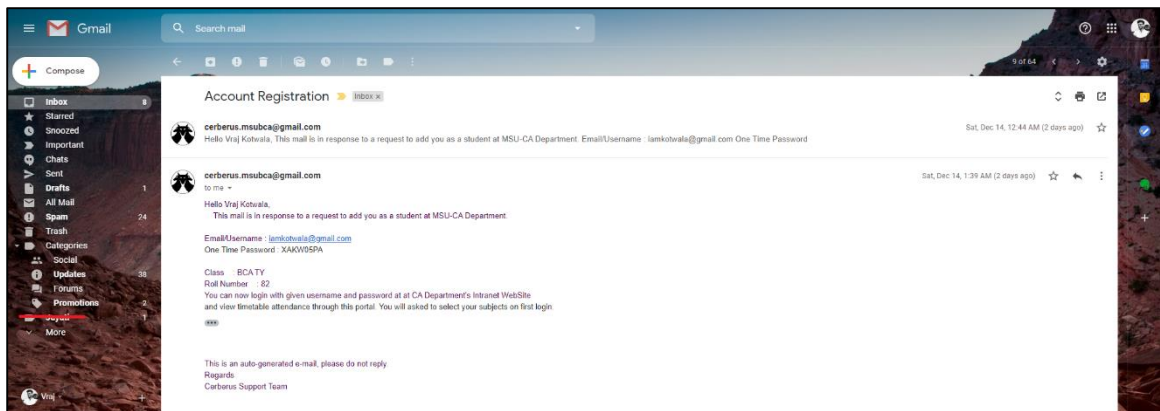


### Logout

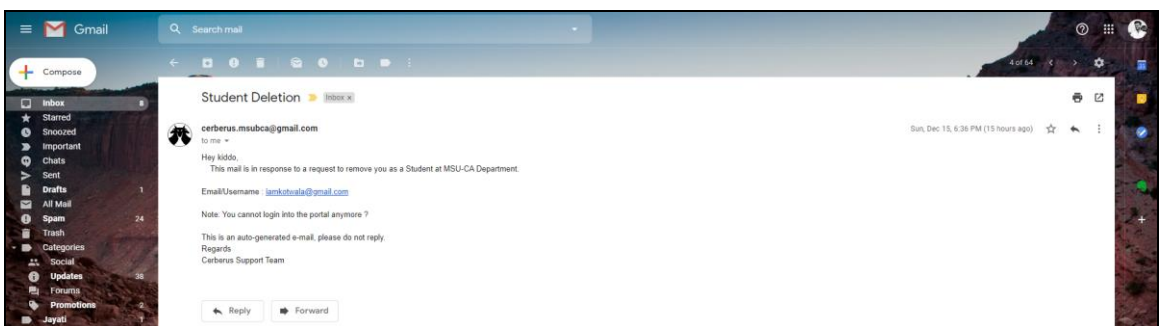
- Verifies before logging out of session.

### Forgot Password

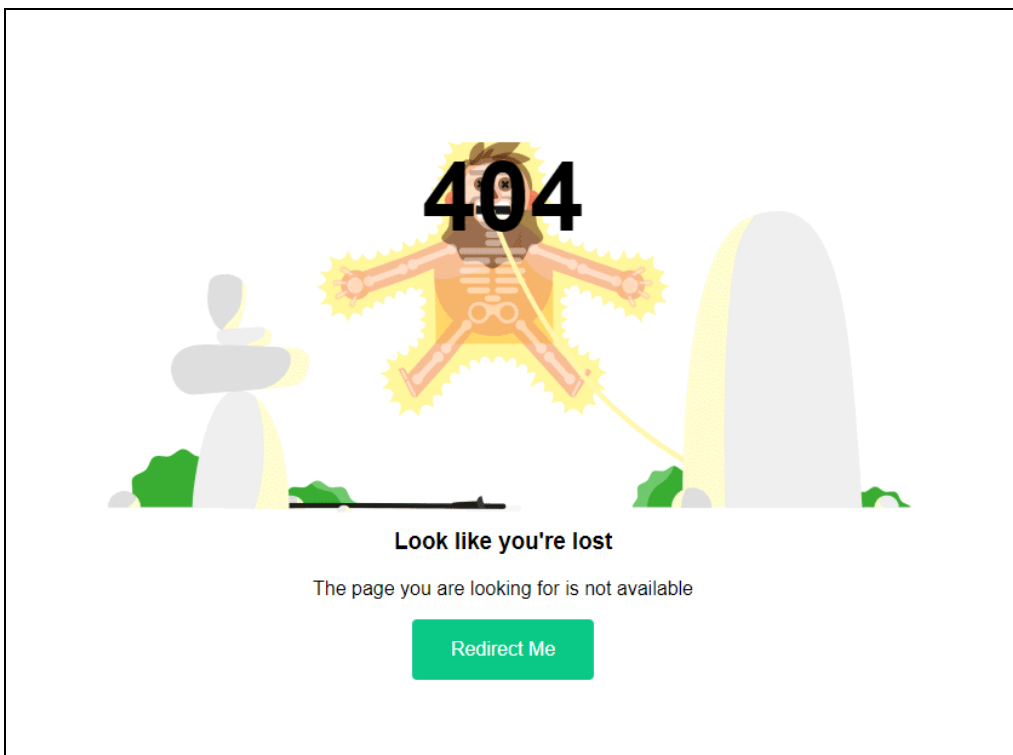
- OTP is sent to mail id of respective student/ Admin.



### Account Registration mail



### Account Deletion Mail



### Error 404 Page

## Testing

Software testing is the process used to identify the correctness, completeness, security and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software.

Test Method:

- a. Unit Testing
- b. Validation Testing
- c. Performance Testing
- d. System Testing
- e. Black box testing

Test Plan:

- Unit testing (RaspberryPi side)

| Test Scenario              | Requirements              | Test cases                               | Test Data                                | Results   |
|----------------------------|---------------------------|--|--|---|
| <b>Enroll</b>              | 3 fingerprint data        | dry, wet, dirty fingerprints             | templates of students and admins         | Enrolment Successful. Unsuccessful only when dirty fingerprints.    |
| <b>Verification</b>        | 3 fingerprint data        | dry, wet, dirty fingerprints             | templates of students and admins         | Verification Successful. Unsuccessful only when dirty fingerprints. |
| <b>Download templates</b>  | 1 finger template         | template with limited size               | templates of students and admins         | templates downloaded successfully                                   |
| <b>upload templates</b>    | 1 finger template         | template with limited size               | templates of students and admins         | templates uploaded successfully                                     |
| <b>get student list</b>    | Start lab session         | Multiple time slots                      | Dummy data in database                   | getting student list successfully                                   |
| <b>get faculty details</b> | Start lab session         | Multiple time slots                      | Dummy data in database                   | getting faculty details successfully                                |
| <b>get time slots</b>      | sync function on power up | 3 times                                  | Dummy data in database                   | Getting time slots successfully                                     |
| <b>get timetable</b>       | sync function on power up | 3 times                                  | Dummy data in database                   | Getting timetable successfully                                      |
| <b>attendance sync</b>     | multiple fingerprint data | fingerprint data on Raspberry's SD cards | fingerprint data on Raspberry's SD cards | Sync data successful.   |

- Unit Testing (Web-Application side)

| Test Scenario         | Requirements  | Test cases  | Test Data  | Results                        |
|-----------------------|---|---|--|--------------------------------|
| <b>Login Module</b>   | All correct combinations entered in login/password Field will let user login. | Check response on entering valid email and password   | iamkotwala@gmail.com as mail. 'admin123' as password | Login Successful               |
|                       | All incorrect combinations entered in login/password field                    | Check response on entering invalid email and password | xyz@gmil.com. "sxasxcw12" as password                | Login Failed                   |
|                       | SQL injections on login form  | SQL inject string                                     | ""'=""   | No harm to database            |
|                       | 5 times login with incorrect password   | 5 times login with incorrect password                 | wrong string of passwords                            | session blocked                |
|                       | teacher and student common login  | login with teachers and students in same form         | Dummy Data stored on different tables                | Login Successful               |
| <b>edit Timetable</b> | select subject to edit Timetable  | check response on save button                         | multiple subjects selected                           | saved timetable successfully   |
|                       | select subject to edit Timetable from past time                               | check response on save button                         | cannot select subjects                               | saved timetable unsuccessfully |
| <b>edit slot</b>      | Time slot required  | check response on save button                         | slot edited  | slot edited successfully       |
| <b>add slot</b>       | time slot required  | check response on save button                         | slot added   | slot added successfully        |
| <b>remove slot</b>    | time slot required  | check response on save button                         | slot deleted   | slot deleted successfully      |
| <b>add student</b>    | student data required   | check response on save button                         | student added with dummy data                        | student added successfully     |
| <b>delete student</b> | student data required   | check response on save button                         | student deleted with dummy data                      | student deleted successfully   |
| <b>add subject</b>    | subjects name required  | check response on save button                         | subject added with dummy data                        | subject added successfully     |

|                       |                        |                               |                                 |                              |
|-----------------------|------------------------|-------------------------------|---------------------------------|------------------------------|
| <b>delete subject</b> | subjects name required | check response on save button | subject deleted with dummy data | subject deleted successfully |
| <b>add faculty</b>    | faculty name required  | check response on save button | faculty added with dummy data   | faculty added successfully   |
| <b>delete faculty</b> | faculty name required  | check response on save button | faculty deleted with dummy data | faculty deleted successfully |

b. Validation Testing

- Using JavaScript and Ajax validated all the forms.
- Validations working correctly? – Yes

c. Performance Testing

- Web application is performing well – Yes
- RaspberryPi is performing well – Yes
- RaspberryPi heating issues – No
- Functions are dependent on each other – Yes

d. System Testing

- Signed in user will be directed directly to the dashboard- Yes
- Proper flow of Web-Application – Yes
- Proper flow of RaspberryPi – Yes
- Functionalities and modules are the same – Yes

e. Black Box Testing

- Independent testing of the system in the department. – Yes

### Bug Fixing/ Test Reports:

During testing phase, we found some bugs in the coding. We had rectified the bugs and re-test the system and it passes the test with different test cases. Some bugs include:

#### *Bugs fixing in Web Application:*

- OTP spam blocking
- Reset password page
- Message.jsp background.
- Fix subject list not showing
- Maximum login attempt fix
- Login query fixed
- First login function fixed
- Session exceptions fixed
- Roll number validation with AJAX
- Username in session variable fixed
- Insert logs at appropriate places
- Hover texts on fontawesome icons
- About Page CSS fixes
- Logout on back button (browser)
- Week year fix in database
- Ajax on subjectID field
- Catch session exceptions on login
- Edit Timetable JavaScript fix

#### *Bugs fixing in RaspberryPi:*

- Exception handling if file not found.
- GPIO pin out
- Database connectivity
- Auto launch shell script
- Loose connectivity
- RTC library
- Keypad library



**Security Features:**

- Prepared-Statement to avoid SQL injections.
- Error page to prevent the server information.
- SHA-256 algorithm used for encryption of sensitive data like passwords.
- Use of Re-captcha in forms to avoid attacks like cross site request forgery attack.
- Use of one-time-password (OTP) to authenticate the user
- Prevention of spamming OTP using time based OPT mailer.
- Session Management to prevent unauthorized access to data tables.
- Menu can be only accessed by teachers/admins with their fingerprint.
- Teacher fingerprint is needed to start lab, prevents illegal attendance marking by students.
- Buzzer to alert teacher of unauthorized access, if any, detected on the system.

**Coding Standard followed:**

- Indentation: Proper and consistent indentation is important in producing easy to read and maintainable programs.
- Inline Comments are used.
- Structured Programming - Structured (or modular) programming techniques are used.
- The names of the classes, subroutines, functions, and methods have verbs in them.
- Variable have meaningful names that convey to a casual observer, the intent of its use. Variables are initialized prior to its first use.
- Proper and consistent scheme for naming (CamelCase and \_underscore)
- Principle of DRY. Don't Repeat Yourself, also known as DIE (duplication is evil). Made classes and functions for repeating blocks of code.
- Proper organization of files and folder.
- Open source code readability.
- Refactoring of code.

## **Limitations and Future Enhancements:**

### Limitations:

- Costly than traditional way of attendance.
- Needs LAN connectivity
- Supervisor needed for smooth operation
- Needs power supply

### Future Enhancements:

- Student Progression – No need to re-register fingerprints when student class is changed.
- Student count in lab – Real-Time.
- In-built fire-safety alarm (Box)

## **Experience and Learning**

Building a whole system from scratch considering all the aspects of software engineering from gathering requirements to planning, designing, proposing the project plan, defining the scope, modeling, analyzing the design, coding, building a database for a whole system, testing was a lifetime experience in itself.

Building an IoT device and a website for a University Dept., working with database, server, interface design, backend coding, assembling everything to get the final output under the guidance of our teachers and the project guide taught us how different is the real world. Working in team made us realize the importance of team spirit, unity and patience.

We faced many difficulties but having a mentor/guide makes everything much easier, every time we were stuck somewhere our guide was there to help us, challenged us to solve the queries on our own and in the end always used to give us the finest solutions.

We learned to solve real problems, learned lot of new techniques and technologies during this period. This project helped us to get out of our comfort zone, learn new technologies and their implementations.

We also learned how to deal with a client, how to elicit requirements from the clients. We learned to design and model the system. For the first time we created test cases for a system made by us and made others test the prototype.

The last 6 months were a roller coaster ride for us consisting of knowledge gain, experience, life lessons, team work, real time dealing. We are grateful to our client and the department for such an amazing experience.

## References and Bibliography:

1. <https://dba.stackexchange.com/questions/52006/database-design-to-track-student-attendance>. - for basic structure of the system database
2. <https://stackoverflow.com/questions/3939764/what-is-a-good-database-design-schema-for-a-attendance-database/3942401#3942401>. - for basic structure of the system database
3. <https://circuitdigest.com/microcontroller-projects/fingerprint-attendance-system-using-arduino-uno>. - basic understanding of working of FPS with Arduino Microcontroller.
4. [https://robu.in/product/sd-card-reading-writing-module-arduino/?gclid=Cj0KCQjwn8\\_mBRCLARIsAKxi0GKwAo3eZptNW\\_0JqorbEiFiDLG7JqhbTHO8yQ2QhS7Gb-Pc-A4DjAaAudhEALw\\_wcB](https://robu.in/product/sd-card-reading-writing-module-arduino/?gclid=Cj0KCQjwn8_mBRCLARIsAKxi0GKwAo3eZptNW_0JqorbEiFiDLG7JqhbTHO8yQ2QhS7Gb-Pc-A4DjAaAudhEALw_wcB) - basic understanding of working of SD CARD with Arduino Microcontroller.
5. [https://robu.in/product/serial-2004-20-x-4-iici2ctwi-blue-backlight-lcd-module/?gclid=CjwKCAjwza\\_mBRBTEiwASDWVvgv5-7Z2ck3JDQexbm28OYU8A0GNDzFmfHsDwZYOSB\\_mKJHKnKkA3RoCr78QAvD\\_BwE](https://robu.in/product/serial-2004-20-x-4-iici2ctwi-blue-backlight-lcd-module/?gclid=CjwKCAjwza_mBRBTEiwASDWVvgv5-7Z2ck3JDQexbm28OYU8A0GNDzFmfHsDwZYOSB_mKJHKnKkA3RoCr78QAvD_BwE) - basic understanding of working of LCD with Arduino Microcontroller.
6. <https://startingelectronics.org/software/arduino/learn-to-program-course/19-serial-input/> - Serial input for Arduino Microcontroller
7. <https://roboindia.com/tutorials/fps-enroll-and-read/> - reading fingerprints
8. <https://www.youtube.com/watch?v=GyprsoB887Y> - reading fingerprints
9. <https://www.youtube.com/watch?v=B1LUp54ysJo> - reading fingerprints
10. <https://drive.google.com/file/d/1mtwhEPZWTWlOP-8fExVauVPzn66eVtkK/view> - FPS datasheets
11. <https://startingelectronics.org/articles/GT-511C3-testing/> - FPS datasheets
12. <https://startingelectronics.org/articles/GT-511C3-fingerprint-scanner-hardware/> - FPS datasheets
13. <https://www.instructables.com/id/Arduino-desktop-application-on-java-in-LAN/> - LAN with microcontroller
14. <https://www.arduino.cc/en/reference/ethernet> - LAN with microcontroller
15. <https://www.rs-online.com/designspark/arduino-networking-basics> - Arduino Networking basics
16. <https://learn.sparkfun.com/tutorials/fingerprint-scanner-hookup-guide#example-code-for-arduino> – FPS Hook-up guide.
17. <https://robu.in/product/4x4-matrix-keypad-membrane-switch-arduino-arm-mcu/> - basic understanding of working of matrix keypad with Arduino Microcontroller.
18. <https://robu.in/product/serial-lcd1602-iici2c-blue-backlight/> - basic understanding of working of Serial LCD with Arduino Microcontroller.
19. <https://robu.in/product/tiny-rtc-real-time-clock-ds1307-i2c-iic-module-for-arduino/> - basic understanding of working of Real-Time-Clock with Arduino Microcontroller.
20. [https://robu.in/product/atmel-mcu-atmega16u2-mega-2560-r3-improved-version-ch340g-cable-arduino-mega-2560-transparent-acrylic-case-arduino-mega-2560/?gclid=Cj0KCQjwMvIBRDmARIsAEoQ8zQRZXfmCU2xr6cv7uVrZ77PqEXLFCOEKLTSkX0KvOt5glzpWTEVXW4aAsJrEALw\\_wcB](https://robu.in/product/atmel-mcu-atmega16u2-mega-2560-r3-improved-version-ch340g-cable-arduino-mega-2560-transparent-acrylic-case-arduino-mega-2560/?gclid=Cj0KCQjwMvIBRDmARIsAEoQ8zQRZXfmCU2xr6cv7uVrZ77PqEXLFCOEKLTSkX0KvOt5glzpWTEVXW4aAsJrEALw_wcB) - basic understanding of working of Arduino Microcontroller.
21. <https://robu.in/product/ethernet-module-enc28j60/> - basic understanding of working of Ethernet with Arduino Microcontroller.
22. <https://w3layouts.com/> - For webpage Layouts
23. <https://forum.arduino.cc/index.php?topic=617118.msg4182544#msg4182544> – SD CARD module initialization
24. <https://stackoverflow.com/questions/4691132/how-to-run-a-background-task-in-a-servlet-based-web-application> - Background task in Java Servlet

25. <https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ScheduledExecutorService.html> - Schedule task in java servlet
26. <https://www.deadcoderising.com/execute-code-on-webapp-startup-and-shutdown-using-servletcontextlistener/> - execute code on webApp startup
27. <https://dev.mysql.com/doc/refman/5.7/en/create-table-generated-columns.html> - generating columns in SQL
28. <https://colorlib.com/wp/one-page-website-templates/> - Webpage Templates
29. [https://www.flipkart.com/sunrobotics-fingerprint-scanner-detection-uart-ttl-gt-511c3-module/p/itm48qfygy88vrt?pid=ETYF48GQ28QWHJQW&lid=LSSTETYF48GQ28QWHJQWZACLUH&marketplace=FLIPKART&cmpid=content\\_learning-toy\\_8965229628\\_gmc](https://www.flipkart.com/sunrobotics-fingerprint-scanner-detection-uart-ttl-gt-511c3-module/p/itm48qfygy88vrt?pid=ETYF48GQ28QWHJQW&lid=LSSTETYF48GQ28QWHJQWZACLUH&marketplace=FLIPKART&cmpid=content_learning-toy_8965229628_gmc) - FPS Scanner
30. <https://www.youtube.com/watch?v=8jhDvsIcGIY> - Arduino Animations
31. <https://forum.arduino.cc/index.php?topic=51850.0> - Arduino Forums
32. <https://circuitdigest.com/microcontroller-projects/arduino-fingerprint-sensor-gt511c3-interfacing> - FPS interfacing with microcontroller
33. <https://dev.to/awwsmm/how-to-encrypt-a-password-in-java-42dh> - SHA2 Encryption with java
34. <https://www.pngkit.com/bigpic/u2w7o0i1o0i1o0w7/> = Cerberus LOGO design idea
35. <https://www.pngkit.com/bigpic/u2w7o0i1o0i1o0w7/> =
36. <https://stackoverflow.com/questions/15378133/could-not-connect-to-smtp-host-smtp-gmail-com-port-465-response-1> - Mailer in java
37. <https://github.com/Cerberus-Biometric> GitHub for Cerberus
38. <https://github.com/Cerberus-Biometric/cerberus-rpi> - GitHub for Cerberus RaspberryPi
39. <https://github.com/Cerberus-Biometric/cerberus-web> - GitHub for Cerberus WebApp
40. <https://raspberrypi.stackexchange.com/questions/44176/connection-guide-for-gt-511c3-fingerprint-scanner-and-raspberry-pi-2-model-b> - FPS with RaspberryPi
41. <https://github.com/zafartahirov/fingerpi> - Open Source code for FPS connection
42. <http://www.geeks3d.com/20150327/meet-the-raspberry-pi-gpio-connector/> - RaspberryPi connection with FPS
43. [https://www.amazon.in/Technotech-Ethernet-100Mbps-Network-Adapter/dp/B072BCF59F/ref=sr\\_1\\_2?keywords=ethernet+to+micro+usb&qid=1565362824&s=computers&sr=1-2](https://www.amazon.in/Technotech-Ethernet-100Mbps-Network-Adapter/dp/B072BCF59F/ref=sr_1_2?keywords=ethernet+to+micro+usb&qid=1565362824&s=computers&sr=1-2) - Ethernet for microcontrollers
44. <https://www.youtube.com/watch?v=Js5KRZSJ2co> - Adding Ethernet to RaspberryPi with GPIO pins
45. <https://www.instructables.com/id/Super-Cheap-Ethernet-for-the-Raspberry-Pi/> - Ethernet Module for RaspberryPi
46. <https://www.raspberrypi.org/forums/viewtopic.php?t=137222> - RaspberryPi Forums
47. <https://raspberrypi.stackexchange.com/questions/70476/tcp-communication-raspberry-pi-and-pc-via-ethernet> - Raspberry Forums
48. [http://www.bristolwatch.com/rpi/rpi\\_i2c\\_ds1307.htm](http://www.bristolwatch.com/rpi/rpi_i2c_ds1307.htm) - RTC w RaspberryPi I2C Interface
49. <https://www.recantha.co.uk/blog/?p=4849> - Character display on RaspberryPi
50. <https://raspberrypi.stackexchange.com/questions/47671/why-my-program-wont-communicate-through-ttyama0-on-raspbian-jessie> - RaspberryPi communication with tty
51. <https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3-4/> - RaspberryPi communication with TTY
52. <https://github.com/zafartahirov/fingerpi/issues/9> - Python driver for Fingerprint Scanner - TTL (GT-511C3)
53. <https://www.raspberrypi.org/forums/viewtopic.php?t=193324> - Interfacing GT511C3 w RaspberryPi
54. <https://github.com/QuickCorp/pyGT511C3> - Python driver for Fingerprint Scanner - TTL (GT-511C3)

55. <https://www.datacamp.com/community/tutorials/pickle-python-tutorial> - Python Pickle tutorial
56. <https://github.com/zafartahirov/fingerpi/issues/17> - Issue with FPS Enrollment
57. <https://github.com/zafartahirov/fingerpi/issues/18> - Problem by Identification
58. <https://github.com/robertvazan/sourceafis-java> - Fingerprint recognition engine for Java that takes a pair of human fingerprint images and returns their similarity score. Supports efficient 1:N search.
59. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4801596/> - Embedding a fingerprint scanner in a RaspberryPi
60. <https://sourceafis.machinezoo.com/> - Algorithm recognizing human fingerprints
61. <https://fingerprint-module-r305-python-and-mysql.readthedocs.io/en/latest/chapter03.html> - finger template to database
62. <https://github.com/kjanko/python-fingerprint-recognition> - finger recognition in python
63. <https://robu.in/product/raspberry-pi-zero-w-v1-3-development-board/> - RaspberryPi zero board.
64. <https://www.youtube.com/watch?v=YaTeTCEgk4> – RTC memory EEPROM
65. <https://github.com/brianrho/FPM> - Library for optical fingerprint sensors
66. <http://ozzmaker.com/raspberry-pi-with-a-3-2-tft-with-touch-control-part-2/> - RaspberryPi with a TFT touch control.
67. [https://www.banggood.in/2-8-Inch-ILI9341-240x320-SPI-TFT-LCD-Display-Touch-Panel-SPI-Serial-Port-Module-p-1206782.html?rmmds=buy&cur\\_warehouse=CN](https://www.banggood.in/2-8-Inch-ILI9341-240x320-SPI-TFT-LCD-Display-Touch-Panel-SPI-Serial-Port-Module-p-1206782.html?rmmds=buy&cur_warehouse=CN) – Touch Panel SPI Serial Port Module
68. <https://www.youtube.com/watch?v=sZVYzEHLcfM> – Using android device as RaspberryPi Monitor
69. <https://www.youtube.com/watch?v=3vuK5w8o8dg> - TFT display on RaspberryPi
70. [https://www.banggood.in/5V-2A-EU-Power-Supply-Micro-USB-AC-Adapter-Charger-For-Raspberry-Pi-p-949349.html?rmmds=search&cur\\_warehouse=CN](https://www.banggood.in/5V-2A-EU-Power-Supply-Micro-USB-AC-Adapter-Charger-For-Raspberry-Pi-p-949349.html?rmmds=search&cur_warehouse=CN)
71. [https://www.banggood.in/Infrared-IR-Wireless-Remote-Controller-Module-Kits-DIY-Kit-HX1838-for-Arduino-Raspberry-Pi-p-1543200.html?rmmds=search&cur\\_warehouse=CN](https://www.banggood.in/Infrared-IR-Wireless-Remote-Controller-Module-Kits-DIY-Kit-HX1838-for-Arduino-Raspberry-Pi-p-1543200.html?rmmds=search&cur_warehouse=CN) – Micro USB AC Adapter charger for RaspberryPi
72. <https://github.com/juj/fbcp-ili9341> - fast display driver for SPI-based LCD displays for Raspberry Pi A, B, 2, 3 and Zero
73. [https://github.com/nopnop2002/ili9340spi\\_rpi](https://github.com/nopnop2002/ili9340spi_rpi) - SPI TFT Library & XPT2046 Touch Screen Library for Raspberry
74. <https://github.com/notro/fbtf/wiki/Framebuffer-use> - framebuffer drivers. All commands are issued from a SSH session.
75. <https://www.adafruit.com/product/1770>
76. <https://www.youtube.com/watch?v=I0TMbUpa4ig> - LCD display on a Raspberry Pi
77. [https://github.com/BLAvery/lib\\_tft24T](https://github.com/BLAvery/lib_tft24T) - Python driver for touchscreen
78. <https://www.youtube.com/watch?v=3zf8irteAdk> - Color touchscreen for Arduino
79. <http://www.spotpear.com/learn/EN/raspberry-pi/Raspberry-Pi-LCD/Install-the-image-to-your-raspberry-pi.html> - install the image to your RaspberryPi
80. <http://www.spotpear.com/learn/EN/raspberry-pi/Raspberry-Pi-LCD/Drive-the-LCD.html> - Drive the LCD
81. <https://raspberrypi.stackexchange.com/questions/34280/white-screen-on-touchscreen-display> - Issue white screen on touchscreen display
82. <https://kodejava.org/how-do-i-count-number-of-online-users/> - Count number of online users
83. <http://www.hubberspot.com/2013/09/how-to-determine-active-users-sessions.html> - session - determine active users.