

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ ط

ترجمہ: اللہ کے نام سے شروع جو بہت مہربان
رحمت والا۔



STUDENT MANAGEMENT SYSTEM



Submitted to: Sir Nayyer Javed

Submitted by: 403-2020-R2
405-2020-R2
409-2020-R2
415-2020-R2
419-2020-R2
426-2020-R2

Final Year Project Report

PROJECT STUDENT MANAGEMENT SYSTEM

By

Talha Sarwar ID: 403-2020-R2,
Asad Muzamil Jutt ID: 405-2020-R2,
Abdullah Hussain ID: 409-2020-R2,
Fazal Abbas ID: 415-2020-R2,
Abdullah Ahmad ID: 419-2020-R2,
Rizwan Fiaz ID: 426-2020-R2,

This Report Presented in Partial Fulfilment of the Requirements for the Diploma of
Associate Engineer in Computer Information Technology **Supervised By**

Mr. Nayyer Javed

Lecturer of Department of Computer Information Technology



**Technical Education and Vocational
Training Authority TEVTA
GCT (GOVERNMENT COLLEGE OF TECHNOLOGY LAHORE)**

APPROVAL

Project titled "Student Management System". Submitted by Abdullah Ahmad, Talha Sarwar, Rizwan Fiaz .to the Department Computer Information Technology, GCT (GOVERNMENT COLLEGE OF TECHNOLOGY LAHORE), has been accepted as satisfactory for the partial Fulfilment of the Requirements for the Diploma of Associate Engineer in Computer Information Technology (CIT) and approved as to its style and contents. The presentation has been held on march 2023.

DECLARATION

I hereby declare that, this project has been done by me under the supervision of Mr. Nayyer Javed Lecturer, of Department Computer Information Technology, GCT (GOVERNMENT COLLEGE OF TECHNOLOGY LAHORE). I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised By: Mr. Nayyer Javed

Lecturer, of Department Computer Information Technology, GCT (GOVERNMENT COLLEGE OF TECHNOLOGY LAHORE).

Submitted By: Abdullah Ahmad, Talha Sarwar, Rizwan Fiaz, Asad Muzamil Jutt, Abdullah Hussain, Fazal Abbas.

Student Management System

Id: 403-2020-R2

Id: 405-2020-R2

Id: 409-2020-R2

Id: 415-2020-R2

Id: 419-2020-R2

Id: 426-2020-R2

Department Computer Information Technology

Acknowledgements

First, I express my heartiest thanks and gratefulness to almighty god for His divine blessing makes us possible to complete the final year project successfully.

I really grateful and wish my profound my indebtedness to Supervisor Mr. Nayyer javed Lecturer, of Department Computer Information Technology, GCT (GOVERNMENT COLLEGE OF TECHNOLOGY LAHORE). Deep knowledge and keen interest of my supervisor in the field of “Game Development” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice has made it possible to complete this project.

I would like to express my heartiest gratitude to Mr. Nayyer javed and HOD CIT, for this kind help to finish our project and to other faculty member and the staff of CIT department of GCT (GOVERNMENT COLLEGE OF TECHNOLOGY WAHDAT ROAD LAHORE).

I would like to thank my entire course mate in GOVERNMENT COLLEGE OF TECHNOLOGY WAHDAT ROAD LAHORE, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

TABLE OF CONTENTS

CHAPTER NO	PAGE NO
1. INTRODUCTION	
1.1 INTRODUCTION	09
2. OBJECTIVE09
2. SYSTEM ANALYSIS	
2.1 INTRODUCTION	10
2.2 EXISTING SYSTEM	11
2.3 PROPOSED SYSTEM	...11
2.4 FEASIBILITY STUDY	...11
2.5 Operational Feasibility	12
3. SYSTEM SPECIFICATION	
3.1 HARDWARE REQUIREMENTS (Minimum Requirement)	13
3.2 SOFTWARE REQUIREMENTS (minimum Requirement)	...13
4. Creating Student Management System	14
By to ways	
4.1 Student Management System with webcam	15
4.1.1 Webcam	
4.1.2 Student Management System Software	18
Worked	
4.2 Student Management System by using Arduino Uno with Esp32 cam	38
4.2.1 Student Management System software intro	...38

4.2.2 Benefits	...38
4.2.3 Components	40
• Male to female	41
• Male to male jumper wire	42
• Bread board	43
• Vero board	44
• Esp32 cam	45
• Arduino uno	45
• Ftdi driver	47
• Potential meter	47
• Header pins	48
• LCD	48
4.2.4 Working	51
Connecting LCD to Arduino uno	...51
• Hardware Required	52
• Circuit	...52
• Schematic	54
• Hello World Example 54
Connecting Esp32 cam, ftdi driver to Arduino uno	57
• Program ESP32-CAM	...57
• Brownout detector or Guru meditation error	60
• Board at COMX is not available – COM Port not selected	61
4.2.5 Student Management System Software	62
Failed	82

Introduction

CHAPTER 1

1.1 Introduction:

A student attendance management system **helps you eliminate paper-based operations completely from the attendance marking and reporting process**. It promotes digital attendance marking and offers solutions for automatically capturing attendance from RFID cards, biometric systems, QR codes, and mobile applications.

1.2 OBJECTIVE:

“Student Management System” is software developed for Maintaining the attendance of the student on the daily basis in the collage. Here the staffs, who are handling the subjects, will be responsible to mark the attendance of the students. Each staff will be given with a separate username and password based on the subject they handle. An accurate report based on the student attendance is generated here. This system will also help in evaluating attendance eligibility criteria of a student. Report of the student’s attendance on weekly and monthly basis is generated.

CHAPTER 2 SYSTEM ANALYSIS:-

INTRODUCTION

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

The various tasks in the system analysis include the following.

- Understanding application.
- Planning.
- Scheduling.
- Developing candidate solution.
- Performing trade studies.
- Performing cost benefit analysis.
- Recommending alternative solutions. ➤ Selling of the system.
- Supervising, installing and maintaining the system.

This system manages to the analysis of the report creation and develops manual entry of the student attendance. First design the students entry form , staff allocation and time table allocation forms. This project will helps the attendance system for the department calculate percentage and reports for eligibility criteria of examination .The application attendance entry system will provide flexible report for all students.

EXISTING SYSTEM

The Existing system is a manual entry for the students. Here the attendance will be carried out in the hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers.

This application requires correct feed on input into the respective field. Suppose the wrong inputs are entered, the application resist to work. so the user find it difficult to use.

PROPOSED SYSTEM:

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the student's attendance. The system provides with the best user interface.

The efficient reports can be generated by using this proposed system.

FEASIBILITY STUDY:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at

this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal.

Operational Feasibility:

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS (Minimum Requirement)

- **Minimum RAM:-**1GB
- **Hard Disk:-**128 GB
- **Processor:-**Intel Pentium 4(1.50 GHZ) or above

3.2 SOFTWARE REQUIREMENTS (minimum Requirement)

- **Operating system :**Windows 10
- **Front_Design:**VB.Net version 10.0 ,.NET framework 4.0
- **Front-End Language :**Visual basic
- **Back-End :** Oracle 10g
- **Back-End Connectivity:**ADO.net

Creating Student Management System By two ways:-

- Student Management System with webcam
- Student Management System by using Arduino Uno with Esp32 cam

4.1 STUDENT MANAGEMENT SYSTEM with webcam

4.1.1 Webcam:

A digital camera that can be linked to a personal computer and used to transmit live video in real time is called a webcam. Webcams are used for **online meetings, conferencing, and online education** the vast majority of the time.

In this project, we can use webcam as a scanner.

Student came front of camera put the student card at the front of camera. The camera Detect the the bar code on the student card.

Using a webcam as a barcode scanner and processing the captured barcode images using image processing techniques can be a powerful and efficient method for barcode recognition. This approach combines the convenience of webcam input with the capabilities of image processing algorithms to extract barcode information.

The process involves the following steps:

Webcam Input: A webcam is used to capture images or video frames that contain barcodes. The webcam serves as the input device, allowing realtime barcode scanning without the need for external hardware scanners.

Image Capture: The webcam captures frames containing the barcode. These frames can be individual images or a continuous video stream, depending on the requirements of the application.

Image Preprocessing: Preprocessing techniques are applied to the captured frames to enhance the quality and clarity of the barcode image. This may include operations such as image resizing, noise reduction, contrast adjustment, and image filtering to improve barcode readability.

Barcode Detection: Image processing algorithms are employed to locate the barcode region within the captured frames. Techniques like edge detection, thresholding, and contour analysis are commonly used to identify the barcode's rectangular boundaries or other distinctive features.

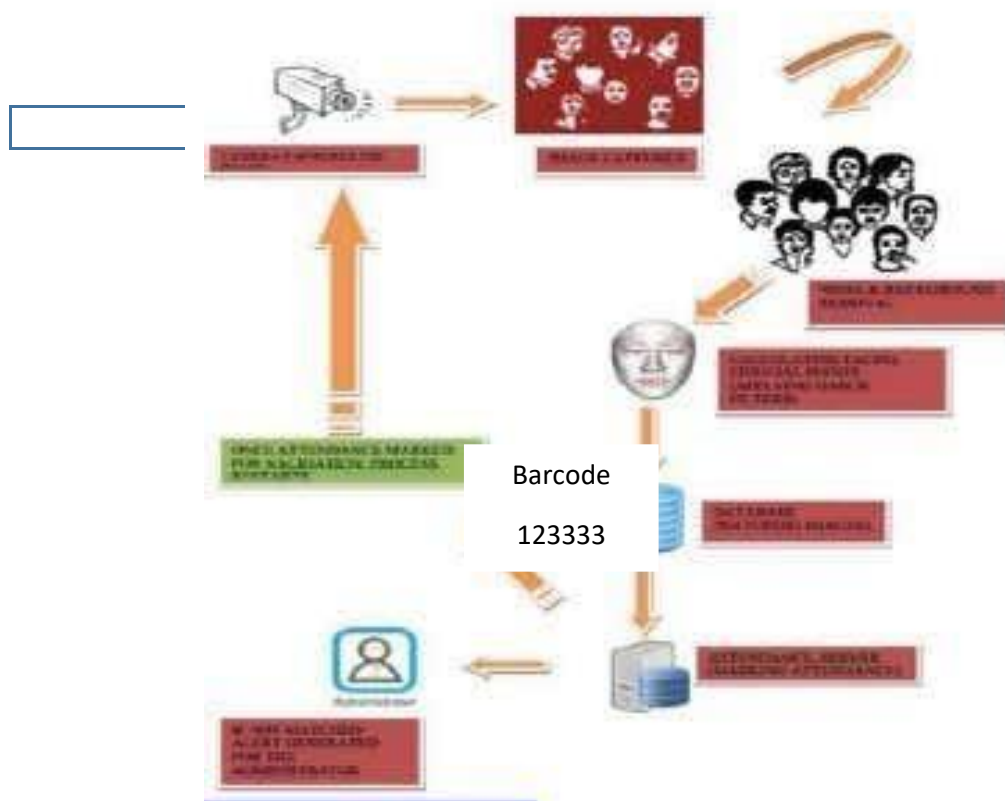
Barcode Decoding: Once the barcode region is detected, a barcode decoding algorithm is applied to extract the encoded information. This algorithm interprets the pattern of bars and spaces in the barcode, decoding it into a meaningful data format such as numeric or alphanumeric values. Popular barcode decoding techniques include 1D barcode decoding (e.g., UPC, EAN) and 2D barcode decoding (e.g., QR code, Data Matrix).

Data Processing and Integration: The decoded barcode information can be further processed, validated, or used for various purposes depending on the application requirements. This data can be integrated into a larger system, database, or used for specific actions or operations.

By utilizing a webcam as a barcode scanner and leveraging imageprocessing techniques, this approach eliminates the need for dedicated barcode scanners and allows for versatile and flexible barcode recognition. It enables real-time scanning, provides a cost-effective solution, and leverages the power of image processing algorithms to accurately extract barcode data from captured frames. This method finds applications in various domains such as inventory management, point-ofsale systems, ticketing, access control, and more.

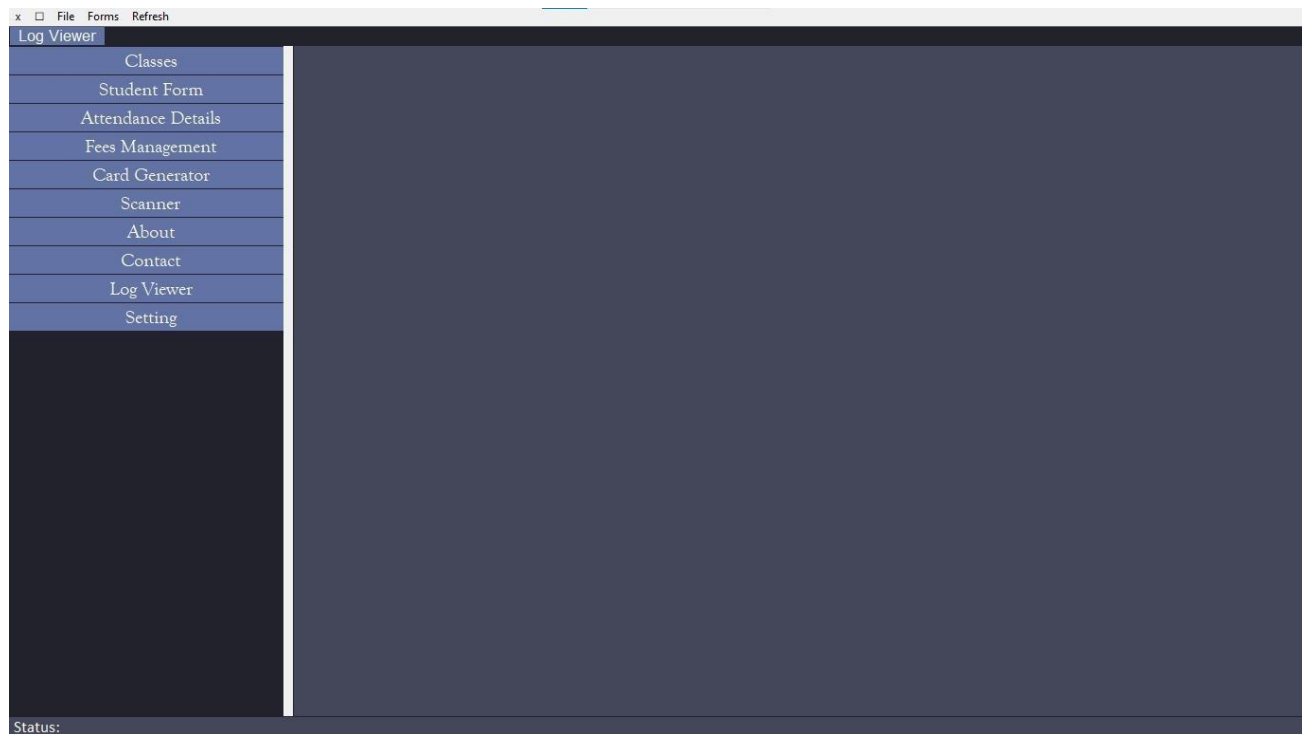
We use it for student card detection for attendance record.

In addition, we EAN barcode decoding method for barcode.



Student Management System Software:

Software functions



Student management:-

The screenshot displays a web-based application window titled "Log Viewer". On the left, there is a form for managing student data. The form includes fields for ID (pre-filled with 1002), Roll No, Student Name, Father Name, Phone Number, Code, Technology, Class, and Shift. A note specifies the Roll No format as "Class-Year-RollNo". Below the form are buttons for Add, Update, Delete, Clear, Import, and Export. On the right, a table lists existing students with columns for ID, Roll No, Student Name, Father Name, Phone Number, Code, Technology, Class, and Shift. The table contains one entry for a student with ID 1001.

ID	Roll No	Student Name	Father Name	Phone Number	Code	Technology	Class	Shift
1001	TCIT-R2-2020-419	Abdullah Ahmad	Irshad Ahmad	0315 0490481	103019930131	Computer Informat	Third Year	R2

A student management form is a graphical user interface (GUI) tool that allows users to manage student information. This form typically includes fields such as student ID, roll number, student name, father's name, phone number, code, technology, class year, shift, and a generated barcode. The form provides functionalities for adding, updating, deleting, importing, exporting, clearing data, and generating barcodes, with support for multiple databases such as SQLite and MySQL.

Student Management System

The screenshot shows a web application window titled "Log Viewer". On the left, there is a form with input fields for "ID" (containing "1002"), "Date", "Student Roll Number", "Student Name", "Technology", "Class", "Shift", "Total Fees", and "Submitted Fees". Below the form are buttons for "Add", "Update", "Delete", "Clear", "Import", and "Export". On the right, there is a table with a "Select" dropdown above it. The table has columns: ID, Date, Student Roll Num, Student Name, Technology, Class, Shift, Total Fees, and Submitted Fees. The first row of data shows: 1001, 06-05-2023, TCIT-R2-2020-419, Abdullah Ahmad, Computer Informat, Third Year, R2, 2100, 2100.

ID	Date	Student Roll Num	Student Name	Technology	Class	Shift	Total Fees	Submitted Fees
1001	06-05-2023	TCIT-R2-2020-419	Abdullah Ahmad	Computer Informat	Third Year	R2	2100	2100

The add function allows users to input a new student record into the database. Users can enter details such as the student's roll number, name, father's name, phone number, code, technology, class year, and shift. Upon adding a new student, the form generates a unique barcode for that student.

The update function enables users to modify an existing student record. Users can make changes to fields like the student's name, father's name, phone number, code, technology, class year, shift, or even generate a new barcode for the student if needed.

The delete function permits users to remove a student record from the database entirely, along with the associated barcode.

The import function allows users to import student data in bulk from an external file, such as a CSV file, while generating barcodes for each imported student.

The export function allows users to save student data, including the generated barcodes, from the database to an external file for further analysis, reporting, or sharing.

The clear function is used to reset the student management form, removing all data and returning it to its default state, including the barcodes.

The Barcode Generate Function Is Used To Generate A Good Barcode for Student According To Requirement.

The student management form is designed to be compatible with multiple database systems, including SQLite and MySQL. SQLite is a lightweight, file-based database system suitable for smaller-scale applications, while MySQL is a robust, server-based database system capable of handling larger datasets and concurrent users. The form can be customized to support one or both of these database systems, providing flexibility and compatibility.

In summary, a student management form with add, update, delete, import, export, clear functions, and barcode generation capabilities provides a user-friendly interface to efficiently manage student information. It simplifies the process of recording and manipulating

student data, supports different database systems, and streamlines data import, export, and barcode generation operations.

Class management:-

A classes form is a graphical user interface (GUI) that allows users to manage classes information. This form typically includes fields for entering data such as the class ID, technology, class year, shift, and description. The form provides options for adding, updating, deleting, importing, exporting, and clearing data from the database.

The add function allows users to enter a new record into the database. In this case, users will enter information about a new class. The update function allows users to modify an existing record in the database. Users can change information such as the technology, class year, or shift for a specific class. The delete function allows users to remove a record from the database entirely.

The import function allows users to add data to the database in bulk by importing data from an external file, such as a spreadsheet or a CSV file. The export function allows users to save data from the database to an external file for use in other applications.

The clear function removes all data from the form and resets it to its default state. This can be useful when starting a new session or when needing to remove all data from the form.

The classes form can be designed to work with multiple database systems, such as SQLite and MySQL. SQLite is a lightweight, file-based database system that is often used for small-scale projects, while MySQL is a more robust, server-based database system that can handle larger

datasets and concurrent users. The form can be designed to support either or both of these database systems.

Overall, a classes form with add, update, delete, import, export, and clear functions can be a useful tool for managing class information in a user-friendly way. It can simplify data entry, retrieval, and modification, and allow users to work with multiple databases.

The screenshot shows a web application window titled 'Log Viewer'. On the left is a sidebar with a 'Log Viewer' tab and a list of fields: ID, Technology, Class Name, Shift, and Details. The 'ID' field is populated with '1007'. To the right of the sidebar is a main content area with a 'Select' dropdown menu. Below the dropdown is a table with the following data:

ID	Technology	Class Name	Shift	Details
1003	Computer Information Techno	First Year	R1	First Year CIT R1
1004	Computer Information Techno	First Year	R2	First Year CIT R2
1005	Computer Information Techno	Second Year	R1	Second Year CIT R1
1006	Computer Information Techno	Second Year	R2	Second Year CIT R2
1001	Computer Information Techno	Third Year	R1	Third Year CIT R1
1002	Computer Information Techno	Third Year	R2	Third Year CIT R2

At the bottom of the window is a toolbar with buttons: Add, Update, Delete, Clear, Import, and Export.

Felid:

- **Technology**
- **Class**
- **Shift**

Attendance management details

An attendance management form is a graphical user interface (GUI) tool that enables users to manage attendance data. This form typically consists of fields for capturing information such as the attendance ID, date, time, attendance status (in or out), student roll number, student name, technology, class year, and shift. The form provides functionalities for adding, updating, deleting, importing, exporting, and clearing data, with support for multiple databases such as SQLite and MySQL.

The add function allows users to input a new attendance record into the database. Users can enter details such as the date, time, attendance status, student roll number, student name, technology, class year, and shift for a particular attendance event. The update function enables users to modify an existing attendance record. Users can make changes to fields like date, time, attendance status, student details, or any other relevant information for a specific attendance entry.

The delete function permits users to remove an attendance record from the database entirely. This option can be used to remove any incorrect or unnecessary entries. The import function allows users to import attendance data in bulk from an external file, such as a CSV file. This feature streamlines the process of adding attendance information from various sources into the database. The export function allows users to save attendance data from the database to an external file. This exported data can be used for further analysis or reporting purposes.

The clear function is used to reset the attendance management form, removing all data and returning it to its default state. This functionality can be useful at the beginning of a new session or when starting fresh with a new set of attendance data.

The attendance management form is designed to be compatible with multiple database systems, including SQLite and MySQL. SQLite is a lightweight, file-based database system suitable for smaller-scale applications, while MySQL is a robust, server-based database system capable of handling larger datasets and concurrent users. The form can be tailored to support one or both of these database systems, providing flexibility and compatibility.

In summary, an attendance management form with add, update, delete, import, export, and clear functions provides a user-friendly interface to efficiently manage attendance data. It simplifies the process of recording and manipulating attendance information, supports different database systems, and streamlines data import and export operations.

Fees management

The screenshot shows a 'Log Viewer' application window. On the left, there is a sidebar with input fields for various attributes: ID (1003), Date, Time, Status, Student Roll Number, Student Name, Technology, Class, and Shift. The main area displays a table with the following data:

ID	Date	Time	Status	Student Roll Num	Student Name	Technology	Class	Shift
1001	03-04-2023	11:20	In	TCIT-R2-2020-419	Abdullah Ahmad	Computer Informat	TCIT	R2
1002	17-05-2023	8:00	In	TCIT-R2-2020-419	Abdullah Ahmad	Computer Informat	Third Year	R2

At the bottom of the sidebar, there are buttons for 'Add', 'Update', 'Delete', 'Clear', 'Import', and 'Export'.

A fee management form is a graphical user interface (GUI) tool that enables users to manage fee-related information. This form typically includes fields such as fee ID, date, student roll number, student name, technology, class year, shift, total fee, and submitted fee. The form provides functionalities for adding, updating, deleting, importing from Excel CSV files, exporting to PDF, text, HTML, CSV, Excel XLSX formats, copying to the clipboard, and clearing data. It supports multiple databases such as SQLite and MySQL.

The add function allows users to input a new fee record into the database. Users can enter details such as the fee date, student roll number, student name, technology, class year, shift, total fee, and submitted fee. This function enables the addition of new fee records to track student payments.

The update function allows users to modify an existing fee record. Users can make changes to fields such as the fee date, student details, total fee, or submitted fee for a specific fee record. This function ensures that the fee records are accurate and up to date.

The delete function allows users to remove a fee record from the database. This function is useful when a fee record becomes obsolete or when there are errors in the record.

The import function allows users to import fee data from an Excel CSV file. This feature simplifies the process of adding fee information in bulk from external sources into the database.

The export function enables users to export fee data in various formats, including PDF, text, HTML, CSV, and Excel XLSX. This functionality facilitates data sharing, reporting, and analysis by allowing users to save fee information in different formats that suit their specific needs.

The clear function resets the fee management form, removing all data and returning it to its default state. This function can be used to start fresh or when there is a need to clear all fee data from the form.

The fee management form supports multiple databases, including SQLite and MySQL. SQLite is a lightweight, file-based database system suitable for small-scale applications, while MySQL is a robust, server-based database system capable of handling larger datasets and concurrent users. The form can be customized to work with either or both of these database systems, providing flexibility and compatibility.

In summary, a fee management form with add, update, delete, import from Excel CSV, export to various formats, clear functions, and multiple database support simplifies the management of fee-related information. It provides an intuitive interface for tracking student payments, supports different databases, and offers options for importing, exporting, and clearing fee data to meet diverse user requirements.

Settings

The settings for a student management system can vary depending on the specific requirements and functionalities. Here is a description of the settings you mentioned:

Advance Setting: This setting provides additional configuration options for advanced users. It may include features such as enabling/disabling specific modules, customizing system behavior, setting default values, configuring integration with other systems, and managing user roles and permissions.

Setting	Advance Setting
Remove Temp File After Close	True
Sqlite Database Backup	True
Sqlite Database Backup Every	Saturday
Sqlite Database Location	Default

Card Generator Setting (Student Card Generator): This setting specifically focuses on configuring the generation of student cards. It allows users to define the layout, design, and content of the student cards. Users can set parameters such as the card size, font styles, logo placement, student information to include, barcode integration, and other visual elements related to student identification.

Setting	Card Generator
Column Per Page	2
Row Per Page	4
Output In	PDF

Color Scheme: The color scheme setting allows users to customize the visual appearance of the student management system. Users can select or define color palettes for various elements of the user interface, such as backgrounds, buttons, text, headings, and other components. This setting enables personalization and branding of the system according to the user's preferences or organizational requirements.

Setting		Color Scheme
mainBackgroundColor	#282a36	
mainLightBackgroundColor	#44475a	
mainDarkBackgroundColor	#21222c	
mainForegroundColor	#f8f8f2	
mainButtonBackgroundColor	#6272a4	
mainButtonForegroundColor	#f8f8f2	
mainLabelBackgroundColor	#44475a	
mainLabelForegroundColor	#f8f8f2	
mainTextBackgroundColor	#44475a	
mainTextForegroundColor	#f8f8f2	
mainTextBorderColor	#6272a4	
mainTextHBackgroundColor	#6272a4	
mainComboFieldBackgroundColor	#44475a	
mainDTextBackgroundColor	#6272a4	
mainDTextForegroundColor	#f8f8f2	
mainDTextBorderColor	#f8f8f2	
mainTreeviewBackgroundColor	#44475a	
mainTreeviewForegroundColor	#f8f8f2	
mainTreeviewHBackgroundColor	#6272a4	
mainTreeviewHForegroundColor	#f8f8f2	
mainButtonActiveBackgroundColor	#bd93f9	
Save		Refresh
Clear		Default

Common Setting: Common settings refer to general configurations that apply to the entire student management system. It typically includes options like system language selection, date/time format, default currency, timezone settings, and other global preferences that affect the overall behavior and functionality of the system.

Setting	
Common Setting	
Messagebox After Add	False
After Add	Clear Record
Messagebox After Update	False
After Update	Clear Record
Messagebox After Delete	False
After Delete	Clear Record
Messagebox After Import	True
Messagebox After Export	True
Automatic Refresh After (ms)	1200
Refresh Thread	True
Full Screen	True
Color Scheme	Dracula Theme
Apply	

DB Configuration: The DB (database) configuration setting involves specifying the details required to establish a connection with the database system. Users can input parameters such as the database type (e.g., SQLite or MySQL), server address, port number, username, password, and database name. This setting ensures the student management system can interact with the chosen database and access or store data accurately.

Setting	
Database Configurations	
Database	sqlite
Host	None
User Name	None
Password	None
MySql Database Name	AS_DB
Test Database	Test

Form Setting: The form setting allows users to customize the appearance and behavior of the data entry forms within the student management system. Users can configure elements such as form layouts, field labels, field validations, data formatting options, and other form-related settings. This setting aims to adapt the forms to the specific needs and preferences of the user or organization.

Setting	Form Setting
Search Frame Height	50
Status Frame Height	20
Button and Detail Frame Width	410
Button Frame Height	30
Main Form Button Frame Width	300
Main Form Status Frame Height	20
Y Space In Fields	30
Width Of Fields	220
X Of Fields	170
ID Start With	1001
Main Form Height of Button	30
Main Heading Font	Algerian
Main Font	Calibri
Field Size	13
Field Height	25
Reset Status After 3s	True

Report Setting: The report setting focuses on customizing the generation and presentation of reports within the student management system. Users can define report templates, select data fields to include, specify sorting and filtering criteria, and set formatting options. This setting enables users to generate tailored reports, such as student attendance reports, fee payment summaries, academic performance reports, and more.

Setting	Report Setting
Report Save Automatically	True
Report Save Location	C:/Users/abdul/OneDrive/Desktop/Report
PDF Landscape Reports	<div> <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Delete"/> </div> <div> <!-- Empty list for Landscape Reports --> </div>
PDF Portrait Reports	<div> <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Delete"/> </div> <div> <!-- Empty list for Portrait Reports --> </div>
PDF Report Size	A4
PDF Top Margin	25
PDF Bottom Margin	25
PDF Left Margin	25
PDF Right Margin	25
Open Report After Save	True

Pages Setting: The pages setting pertains to configuring the structure and content of the various pages or screens within the student management system. Users can define the navigation menu, create or edit pages, arrange page layouts, and manage access permissions for different user roles. This setting allows users to organize and customize the system's interface to suit their specific workflow and information requirements.

Overall, these settings provide administrators and users with flexibility and control over various aspects of the student management system, ensuring it can be tailored to meet specific needs, preferences, and organizational requirements.

Card generator

The image displays two screenshots of the Student Card Generator GUI. Both screenshots show a window titled 'Log Viewer' with a menu bar (File, More, Refresh) and a toolbar (Add, Generate, Delete, Clear, Import, Export).

Top Screenshot (Single Student Mode):

- The 'Generate' dropdown is set to 'Single Student'.
- Input fields are provided for: Student Roll Number, Student Name, Technology, Class, Shift, and Code.
- The main area displays a table titled 'STUDENT NAMES' with columns: Roll No, Student Name, Code, Technology, Class, and Shift.

Bottom Screenshot (Multiple Student Mode):

- The 'Generate' dropdown is set to 'Multiple Student'.
- Input fields are provided for: Technology, Class, and Shift.
- The main area displays a table titled 'STUDENT NAMES' with columns: Roll No, Student Name, Code, Technology, Class, and Shift.

The Student Card Generator form is a graphical user interface (GUI) tool that facilitates the creation of student cards based on different criteria

such as technology, class, shift, and individual students. The form provides functionalities for adding to the table for generating cards, deleting from the table, generating cards, clearing the table, importing data, and exporting generated cards. Here's a description of each functionality:

Add to Table for Generating: This function allows users to add students to a table for generating cards. Users can input details such as technology, class, shift, and other relevant student information into the form. The added students are listed in a table, ready for generating their respective cards.

Delete from Generating: This function enables users to remove selected students from the table for generating cards. Users can choose specific entries or use checkboxes to select multiple students and delete them from the list.

Generate: This function generates student cards based on the selected criteria. Users can choose from different options, such as generating cards for all students, generating cards based on technology and class, or generating cards based on technology, class, and shift. The generated cards can include student details, photographs, barcodes, and any other required information.

Clear: The clear function resets the Student Card Generator form, clearing the table and removing all data from the form. This allows users to start fresh or prepare the form for a new set of student card generation.

Import: The import function allows users to import student data from an external source, such as a CSV file. Users can import student details along with their respective technology, class, shift, and other relevant information. This feature simplifies the process of adding multiple students to the card generation table.

Export: The export function enables users to save the generated student cards to an external file or format. Users can export the cards as PDF, image files, or any other preferred format. This functionality facilitates printing or sharing the generated cards with other systems or applications.

The Student Card Generator form provides a user-friendly interface to efficiently manage and generate student cards based on various criteria. It offers flexibility in selecting students for card generation, supports importing and exporting data, and provides options to add, delete, generate, and clear student card data.

Scanner [attendance scanner page]

Our student management system incorporates a barcode scanning functionality that utilizes a webcam as the scanning device. This allows for the efficient and convenient scanning of student barcodes to record attendance in real-time. The process involves capturing an image using the webcam, processing it using image analysis techniques, detecting the student barcode within the image, and subsequently adding attendance records to the student management system's database.

Here is a breakdown of the steps involved:

Webcam Image Capture: The webcam captures an image that includes the area where the barcode is expected to be located. This image serves as the input for further processing.

Image Processing: The captured image undergoes image processing techniques to enhance its quality and facilitate barcode detection. This may involve operations such as resizing, noise reduction, contrast adjustment, and image filtering to optimize barcode readability.

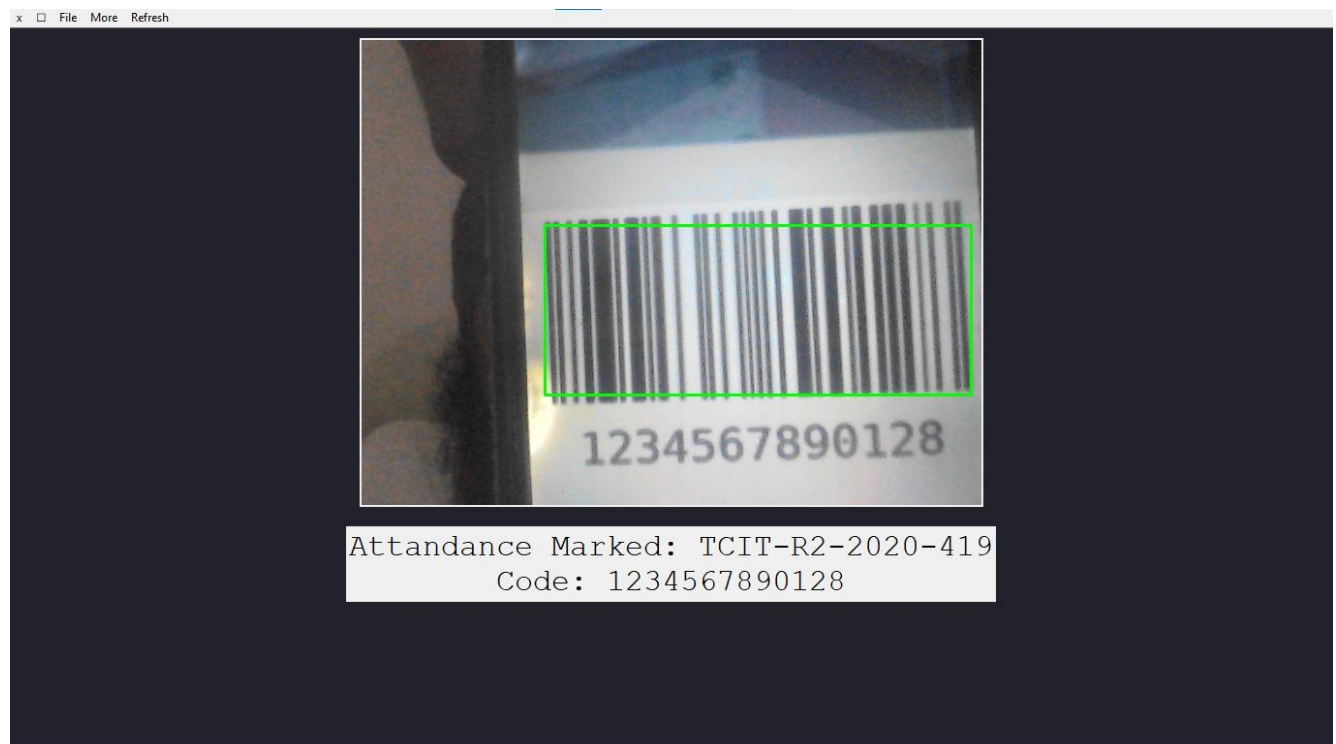
Barcode Detection: Image processing algorithms are employed to identify and locate the student barcode within the captured image. Techniques like edge detection, thresholding, and pattern recognition can be utilized to pinpoint the barcode's position accurately.

Barcode Decoding: Once the barcode is detected, a barcode decoding algorithm is applied to extract the encoded information from the barcode image. This algorithm interprets the pattern of bars and spaces, deciphering the barcode's unique identifier or data.

Attendance Recording: With the decoded barcode information, the student's attendance is recorded in real-time. The student management system's database is accessed, and an attendance record is added for the identified student. This process occurs instantly, providing immediate updates to the student's attendance status.

Database Integration: The attendance record is integrated into the student management system's database, ensuring that accurate attendance information is stored for future reference and analysis. The database can be updated with additional details, such as the date, time, and other relevant information related to the attendance event.

By utilizing a webcam as a barcode scanner and leveraging image processing techniques, this approach streamlines the attendance recording process. It eliminates the need for manual data entry, reducing errors and improving efficiency. The real-time nature of the system enables immediate attendance updates, providing up-to-date information for monitoring and analysis. This barcode scanning functionality enhances the overall functionality of the student management system and contributes to a streamlined and automated attendance management process.



4.2 STUDENT MANAGEMENT SYSTEM By Using Arduino UNO with Esp32 cam

First, we starting about the software, which used in this project:

Student Management System Software Intro:

Student management software is application software for schools and colleges to manage student data. It is an interactive platform for all the student related management of school and colleges. The information can be shared easily with authorized users, records searched and reports generated as well. The software provides a secure database structure with a login authenticated system that organizes stores and retrieves real time information.

Benefits:

- Student management system software help organization to reduce their financial costs by cutting back on the use of paper.
- The system is useful in managing fees related to all type of student record.
- Another feature found in the student management software is for students to use. Through these feature students can check their attendance, grades, test schedule, and class timetable.
- If access is made available, parents can be made aware of their child's academic success in school.
- Parents can benefit from this system as it allows teacher to communicate with the parents about their child's performance or behavior in school.

- They are error free, easy to use and implement that doesn't require any sort of training to be given to the staff. With this any sort of functions related to students be it attendance, grading, admission, change in information etc. It's can be managed very easily.

Major Modules:

- Class management.
- Examination management. □ Fees management.
- Create Student's Card.
- Generate Student Attendance

Components:

- Male to female
- Male to male jumper wire
- Bread board
- Vero board
- Esp32 cam
- Arduino uno
- Ftdi driver
- Potential meter
- Header pins
- LCD

1] Male to female wires

The male to female jumper wires, in particular, **have a male pin on one end and a female connector on the other**, making them suitable for a wide range of projects that require connecting components with different types of pins or connectors.

Function of male to female jumper wires:

These are male to female jumper wires used in **connecting the female header pin of any development board to other development boards having a male connector**. They are simple wires that have connector pins at each end allowing them to be used to connect two points to each other.



2] Male to male jumper wire

Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you will need.

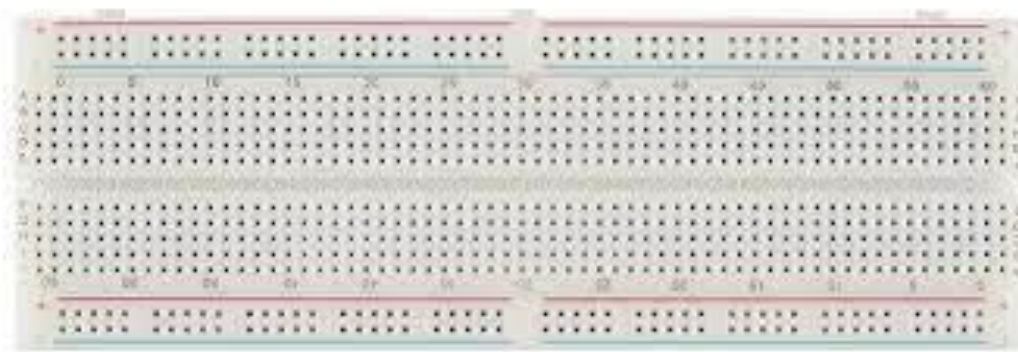
These are male-to-male jumper wires **used in connecting the male header pin of any development board to other development boards having a male connector**. They are simple wires that have connector pins at each end allowing them to be used to connect two points to each other.



3] Bread board

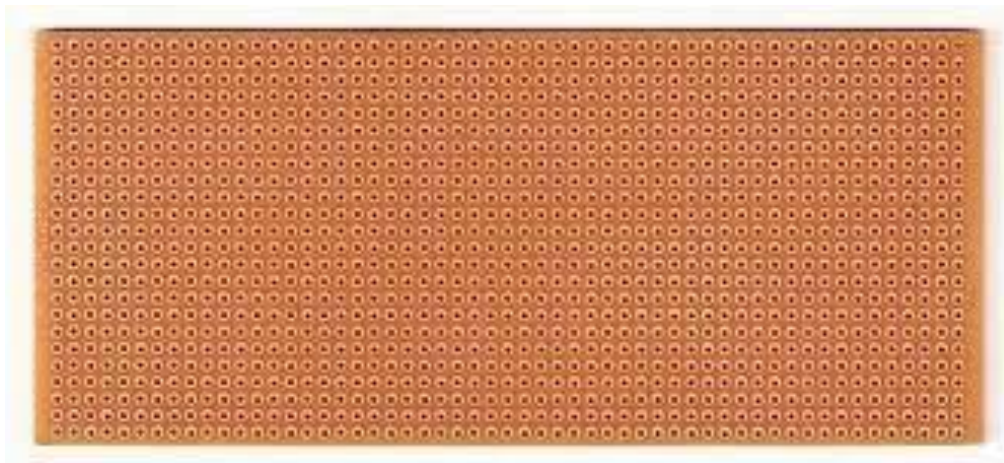
A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a per_board or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable. For this reason, breadboards are also popular with students and in technological education.

A breadboard (sometimes called a plugblock) is used for **building temporary circuits**. It is useful to designers because it allows components to be removed and replaced easily. It is useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit.



4] Vero board

Vero board can be used to **build prototypes for bench testing, design early electronic circuits, or produce entire electronic equipment in small quantities**. The Vero Electronics Department used Vero board for prototype building for the first time in 1961.



5] Esp32 cam

ESP32-CAM is a **WiFi+ bluetooth dual-mode development board that uses PCB onboard antennas and cores based on ESP32 chips**. It can work independently as a minimum system. Ultra-small 802.11b/g/n Wi-Fi + BT/BLE SoC module. Low-power dual-core 32-bit CPU for application processors. With Micro USB to Serial Port CH340. The ESP32-CAM is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera and provides onboard TF card slot. The ESP32-CAM can be widely used in intelligent IoT applications such as **wireless video monitoring, WiFi image upload, QR identification**, and so on.



6] Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller, developed by Arduino.cc, and initially released in 2010. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be **integrated into a variety of electronic projects**. This board can be interfaced with other Arduino boards, Arduino shields, and Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

Features of the Arduino UNO:

- Microcontroller: ATmega328.
- Operating Voltage: 5V.
- Input Voltage (recommended): 7-12V.
- Input Voltage (limits): 6-20V.
- Digital I/O Pins: 14 (of which 6 provide PWM output) □ Analog Input Pins: 6.
- DC Current per I/O Pin: 40 mA.
- DC Current for 3.3V Pin: 50 mA



8] Ftdi driver

FTDI USB Serial Port driver is **the software that helps your operating system to communicate with USB Serial Port devices.**

It develops, manufactures, and supports devices and their related cables and software drivers for converting RS-232 or TTL serial transmissions to and from USB signals, in order to provide support for legacy devices with modern computers.



9] Potential meter

A potentiometer is a **three-terminal resistor with a sliding or rotating** are used, one end and the wiper, it acts as a variable resistor or rheostat. **contact that forms an adjustable voltage divider.** If only two terminals Potentiometer. A typical single-turn potentiometer.



10] Header pins

Pin header connector, a type of connector, English name: Pin Header. This connector is widely used in PCB circuit boards in electronics, electrical appliances, and meters. Its role is to **serve as a bridge between blocked or isolated circuits in the circuit, and carry the current or signal transmission task.**



11] LCD

LCD model: Hitachi HD44780

The HD44780 is a controller for character-based liquid crystal displays (LCDs). In addition to the standard pre-programmed characters, it allows you to define up to eight of your own graphic patterns on a 5-by-7 (or sometimes 5-by-8) grid.

Pin#	Name	Direction	Description
1	Vss	Power	Power Supply Ground
2	Vcc	Power	Power Supply (+3.3 or +5 V depending on module)
3	Vee	Input	Contrast Adjustment (analog input)
4	RS	Input	Register Select (0 = command, 1 = data)
5	R/W	Input	Read/Write (0 = write to display module, 1 = read from display module)
6	E	Input	Clock Enable (falling-edge triggered)
7	DB0	I/O	Data Bit 0 (not used in 4-bit operation)
8	DB1	I/O	Data Bit 1 (not used in 4-bit operation)
9	DB2	I/O	Data Bit 2 (not used in 4-bit operation)
10	DB3	I/O	Data Bit 3 (not used in 4-bit operation)
11	DB4	I/O	Data Bit 4
12	DB5	I/O	Data Bit 5
13	DB6	I/O	Data Bit 6
14	DB7	I/O	Data Bit 7
15	LED+	Power	Backlight Anode (+) (if applicable)
16	LED-	Power	Backlight Cathode (-) (if applicable)



- A **register select (RS)** pin that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.

- A **Read/Write (R/W) pin** that selects reading mode or writing mode
 - An **Enable pin** that enables writing to the registers
- 8 **data pins (D0 -D7)**. The states of these pins (high or low) are the bits that you're writing to a register when you write, or the values you're reading when you read.

There's also a **display contrast pin (Vo)**, **power supply pins (+5V and GND)** and **LED Backlight (BKlt+ and BKlt-)** pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. The [LiquidCrystal Library](#) simplifies this for you so you don't need to know the low-level instructions.

The Hitachi-compatible LCDs can be controlled in two modes: 4-bit or 8-bit. The 4bit mode requires seven I/O pins from the Arduino, while the 8-bit mode requires 11 pins. For displaying text on the screen, you can do most everything in 4-bit mode, so example shows how to control a 16x2 LCD in 4-bit mode.

Working:

Connecting LCD to Arduino uno

Hardware Required

- Arduino Board
- LCD Screen (compatible with Hitachi HD44780 driver)
- pin headers to solder to the LCD display pins
- 10k ohm potentiometer
- 220 ohm resistor
- hook-up wires
- breadboard **Circuit**

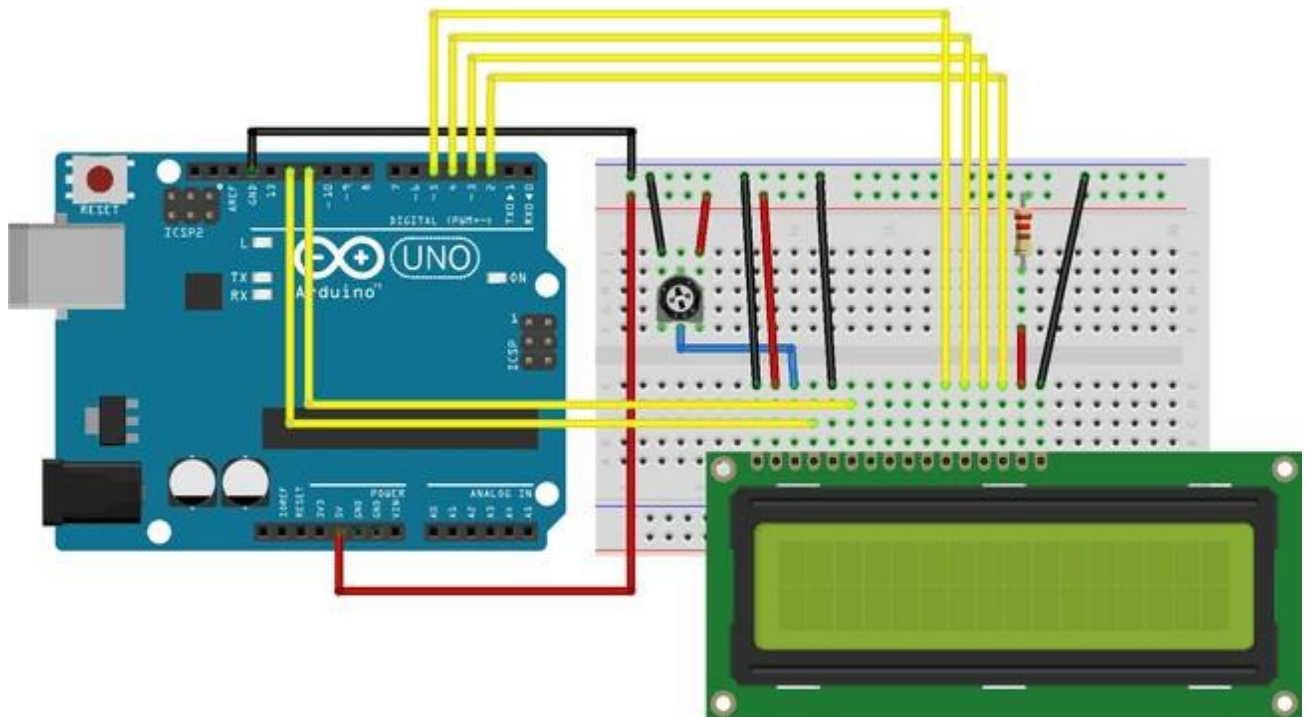
Before wiring the LCD screen to your Arduino board we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image further up.

To wire your LCD screen to your board, connect the following pins:

- LCD RS pin to digital pin 12

- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- LCD R/W pin to GND □ LCD VSS pin to GND
- LCD VCC pin to 5V
- LCD LED+ to 5V through a 220 ohm resistor
- LCD LED- to GND

Additionally, wire a 10k potentiometer to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3).





This sketch prints "Hello World!" to the LCD

Student Management System

and shows the time.

The circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * LCD VSS pin to ground
- * LCD VCC pin to 5V * 10K resistor:
- * ends to +5V and ground
- * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008 by

David A. Mellis library modified 5 Jul

2009 by Limor Fried

(<http://www.ladyada.net>) example

added 9 Jul 2009 by Tom Igoe modified

22 Nov 2010 by Tom Igoe modified 7

Nov 2016 by Arturo Guadalupi

This example code is in the public domain.

<https://docs.arduino.cc/learn/electronics/lcd-displays>

*/

// include the library code:

#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin

Student Management System

```
// with the arduino pin number it is connected to const
int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2); // Print a
  message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

For more testing code available on this site:

<https://docs.arduino.cc/learn/electronics/lcd-displays>

Connecting Esp32 cam, ftdi driver to Arduino uno

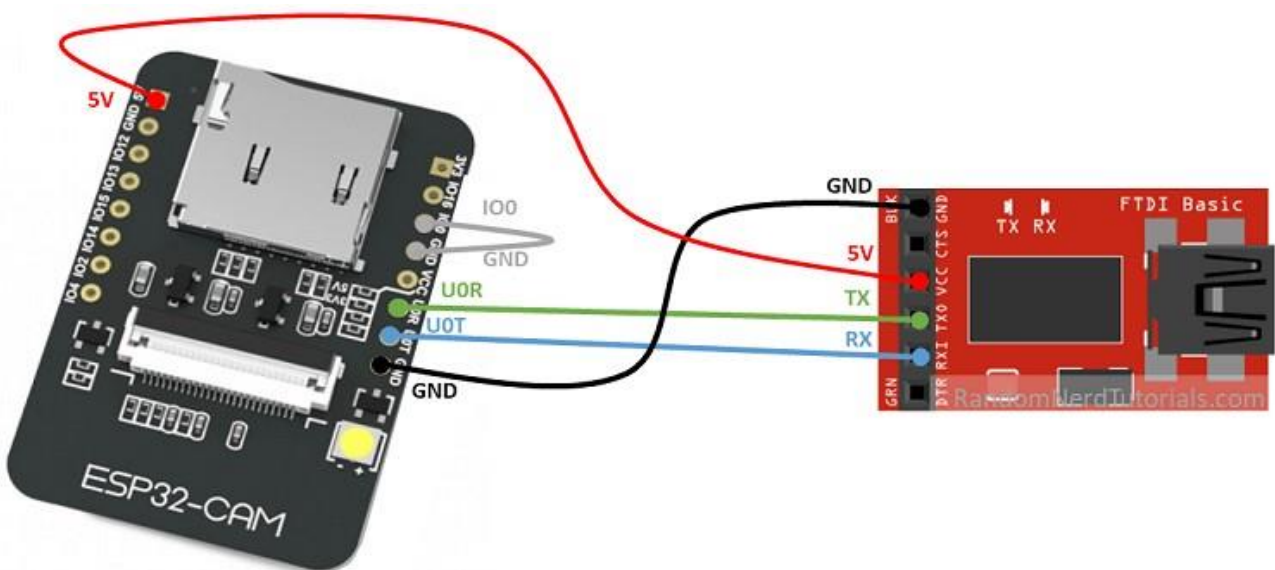
To program the ESP32-CAM board with Arduino IDE, you need to have Arduino IDE installed as well as the ESP32 add-on. Follow the next tutorial to install the ESP32 add-on, if you haven't already:

- [Installing the ESP32 Board in Arduino IDE](#)

Program ESP32-CAM (Upload Code with Arduino IDE)

To upload code to the ESP32-CAM (AI-Thinker) using Arduino IDE, follow the next exact steps.

Connect the ESP32-CAM board to your computer using an [FTDI programmer](#). Follow the next schematic diagram:



Many FTDI programmers have a jumper that allows you to select 3.3V or 5V. Make sure the jumper is in the right place to select 5V.

GPIO 0 needs to be connected to GND so that you're able to upload code.

ESP32-CAM	FTDI Programmer
GND	GND
5V	VCC (5V)
U0R	TX
U0T	RX
GPIO 0	GND

To upload code to the ESP32-CAM using Arduino IDE, follow the next steps:

- 1) Go to **Tools > Board** and select **AI-Thinker ESP32-CAM**. You must have the [ESP32 add-on installed](#). Otherwise, this board won't show up on the Boards menu.
- 2) Go to **Tools > Port** and select the COM port the ESP32-CAM is connected to.
- 3) For demonstration purposes, you can upload a blank sketch to your board:

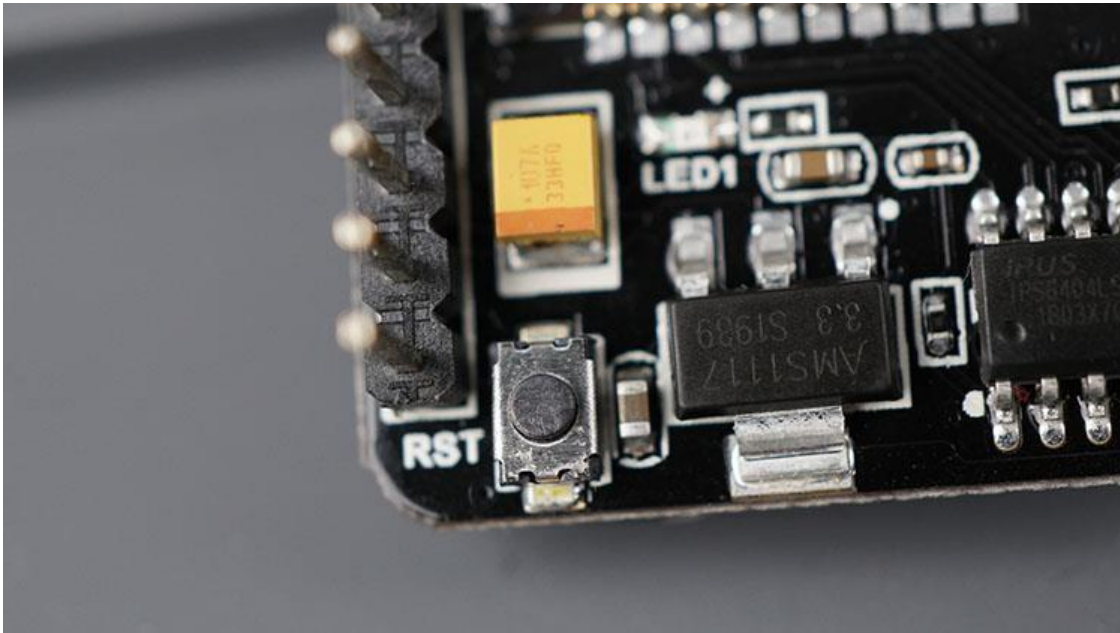
```
void setup() {
  // put your setup code here, to run once:
}
void loop()
{
  // put your main code here, to run repeatedly: }
```

- 4) Then, click the **Upload** button in your Arduino IDE.

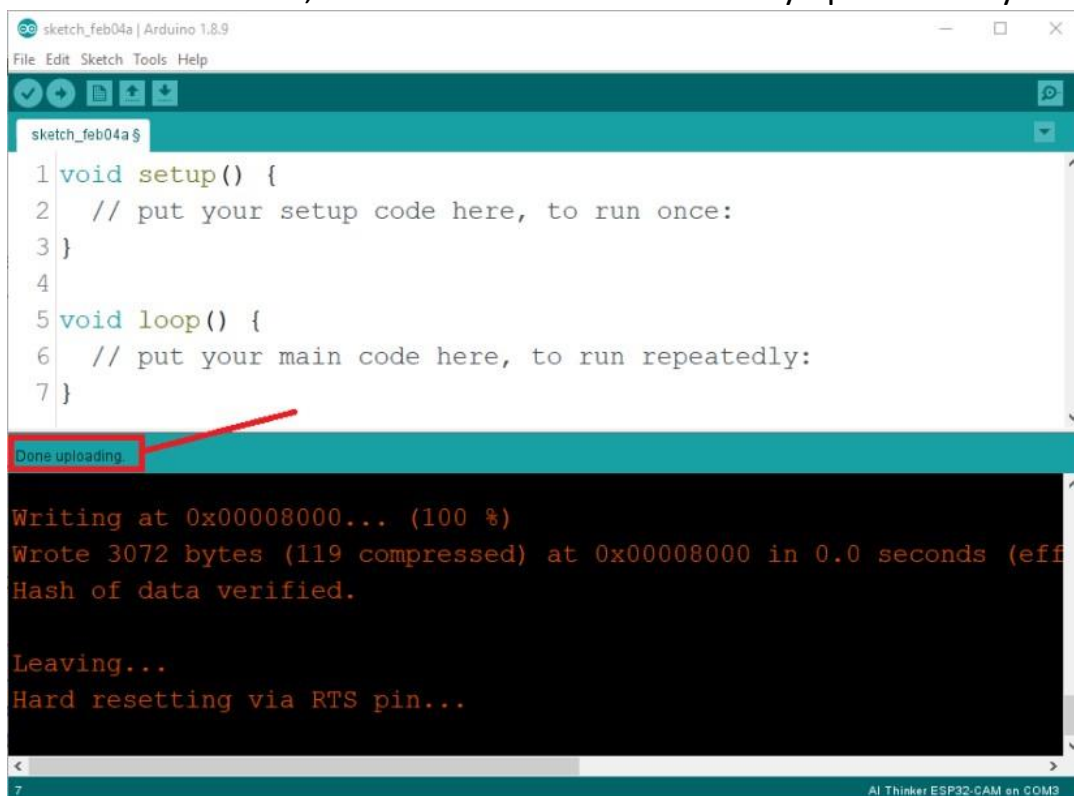


- 5) When you start to see some dots on the debugging window, press the ESP32-CAM on-board RST button.

```
esptool.py v2.6-beta1  
Serial port COM10  
Connecting.....
```



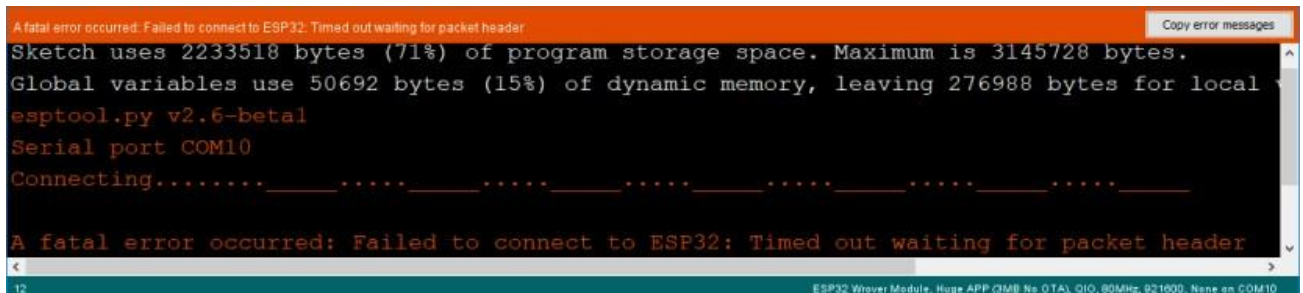
After a few seconds, the code should be successfully uploaded to your board.



```
sketch_feb04a | Arduino 1.8.9  
File Edit Sketch Tools Help  
sketch_feb04a $  
1 void setup() {  
2   // put your setup code here, to run once:  
3 }  
4  
5 void loop() {  
6   // put your main code here, to run repeatedly:  
7 }  
Done uploading.  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds (effective 1152 kbytes/s)  
Hash of data verified.  
Leaving...  
Hard resetting via RTS pin...  
7 AI Thinker ESP32-CAM on COM3
```

- 6) When you see the “**Done uploading**” message, you need to remove GPIO 0 from GND and press the RST button to run your new code.

Failed to connect to ESP32: Timed out waiting for packet header

A screenshot of the Arduino IDE Serial Monitor window. The window has a black background with white and red text. At the top, a red banner reads "A fatal error occurred: Failed to connect to ESP32. Timed out waiting for packet header". Below this, white text shows: "Sketch uses 2233518 bytes (71%) of program storage space. Maximum is 3145728 bytes. Global variables use 50692 bytes (15%) of dynamic memory, leaving 276988 bytes for local variables. esptool.py v2.6-beta1 Serial port COM10 Connecting.....". At the bottom, a red line of text reads "A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header". The status bar at the bottom of the window shows "12" on the left and "ESP32 Wrover Module. Huge APP (MB No OTA), 80MHz, 921600, None on COM10" on the right.

This error means that the ESP32-CAM is not in flashing mode or it is not connected properly to the FTDI programmer.

Brownout detector or Guru meditation error

When you open your Arduino IDE Serial Monitor and the error message “Brownout detector was triggered” is constantly being printed over and over again. It means that there’s some sort of hardware problem.

It’s often related to one of the following issues:

- Poor quality USB cable;
- USB cable is too long;
- Board with some defect (bad solder joints);
- Bad computer USB port;
- Or not enough power provided by the computer USB port.

Solution:

- Try a different shorter USB cable (with data wires);
- Use a different computer USB port or use a USB hub with an external power supply;
- Some readers were using 3.3V and reported that when powering the ESP32CAM with 5V, the issue was fixed.

Board at COMX is not available – COM Port not selected

A screenshot of an Arduino IDE console window. The window has a black background with red and white text. At the top, there is a red banner with white text that reads "the selected serial port Failed to execute script esptool does not exist or your board is not connected". Below this, the text shows a traceback from a Python script: "File 'site-packages\serial__init__.py', line 88, in serial_for_url", "File 'site-packages\serial\serialwin32.py', line 62, in open", and "serial.serialutil.SerialException: could not open port 'COM8': WindowsError(2, 'The system cannot find the file specified.')". The error message "Failed to execute script esptool" is repeated twice. At the bottom of the console, a status bar shows "10" on the left and "ESP32 Wrover Module, Huge APP (3MB No OTA), QIO, 80MHz, 921600, None on COM8" on the right. A "Copy error messages" button is visible in the top right corner of the console area.

If you get the following error or similar:

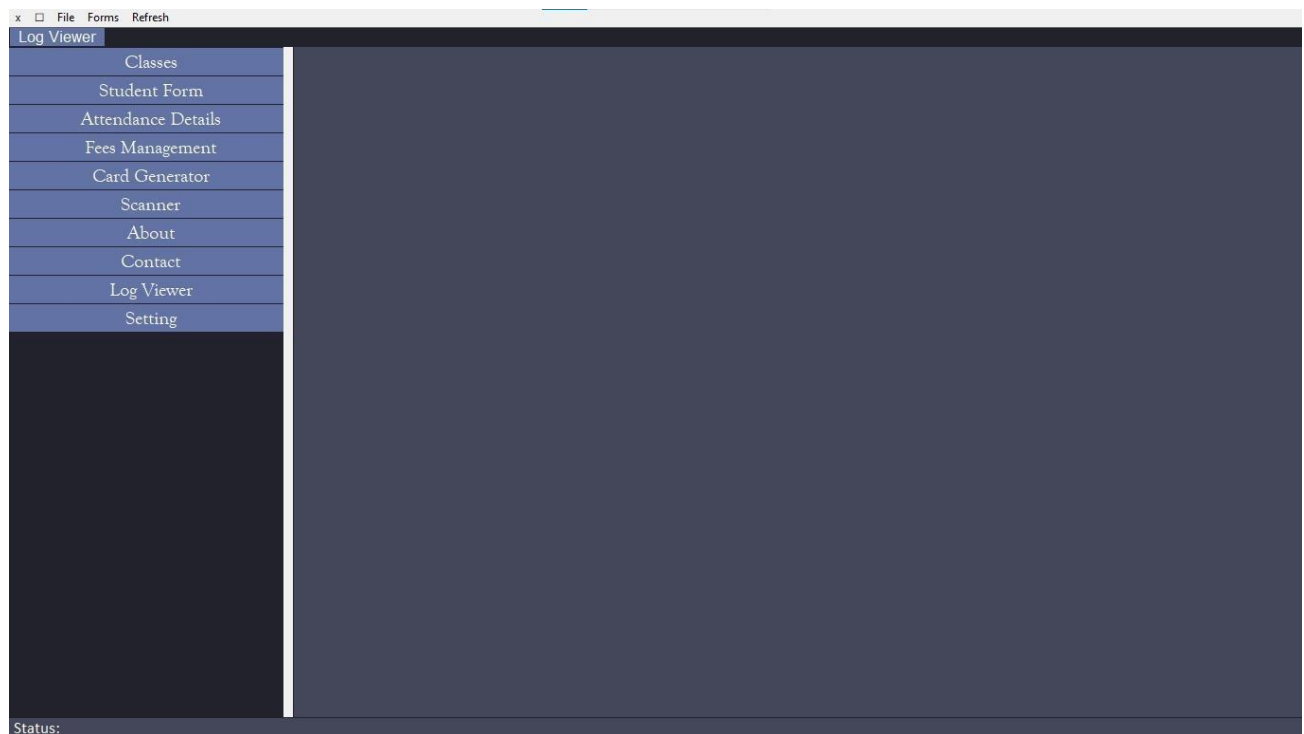
```
serial.serialutil.SerialException: could not open port 'COM8': WindowsError(2,
'The system cannot find the file specified.')
Failed to execute script esptool
The selected serial port Failed to execute script esptool
Does not exist or your board is not connected
Board at COM8 is not available
```

It means that you have not selected the COM port in the Tools menu. In your Arduino IDE, go to **Tools > Port** and select the COM port the ESP32 is connected to.

It might also mean that the ESP32-CAM is not establishing a serial connection with your computer or it is not properly connected to the USB connector

Student Management System Software:

Software functions



Student management:-

ID	Roll No	Student Name	Father Name	Phone Number	Code	Technology	Class	Shift
1001	TCIT-R2-2020-419	Abdullah Ahmad	Irshad Ahmad	0315 0490481	103019930131	Computer Informat	Third Year	R2

A student management form is a graphical user interface (GUI) tool that allows users to manage student information. This form typically includes fields such as student ID, roll number, student name, father's name, phone number, code, technology, class year, shift, and a generated barcode. The form provides functionalities for adding, updating, deleting, importing, exporting, clearing data, and generating barcodes, with support for multiple databases such as SQLite and MySQL.

Student Management System

The screenshot shows a web application window titled "Log Viewer". On the left, there is a form with input fields for "ID" (containing "1002"), "Date", "Student Roll Number", "Student Name", "Technology", "Class", "Shift", "Total Fees", and "Submitted Fees". Below the form are buttons for "Add", "Update", "Delete", "Clear", "Import", and "Export". On the right, there is a table with a "Select" dropdown above it. The table has columns: ID, Date, Student Roll Num, Student Name, Technology, Class, Shift, Total Fees, and Submitted Fees. The first row of data shows: 1001, 06-05-2023, TCIT-R2-2020-419, Abdullah Ahmad, Computer Informat, Third Year, R2, 2100, 2100.

ID	Date	Student Roll Num	Student Name	Technology	Class	Shift	Total Fees	Submitted Fees
1001	06-05-2023	TCIT-R2-2020-419	Abdullah Ahmad	Computer Informat	Third Year	R2	2100	2100

The add function allows users to input a new student record into the database. Users can enter details such as the student's roll number, name, father's name, phone number, code, technology, class year, and shift. Upon adding a new student, the form generates a unique barcode for that student.

The update function enables users to modify an existing student record. Users can make changes to fields like the student's name, father's name, phone number, code, technology, class year, shift, or even generate a new barcode for the student if needed.

The delete function permits users to remove a student record from the database entirely, along with the associated barcode.

The import function allows users to import student data in bulk from an external file, such as a CSV file, while generating barcodes for each imported student.

The export function allows users to save student data, including the generated barcodes, from the database to an external file for further analysis, reporting, or sharing.

The clear function is used to reset the student management form, removing all data and returning it to its default state, including the barcodes.

The Barcode Generate Function Is Used To Generate A Good Barcode for Student According To Requirement.

The student management form is designed to be compatible with multiple database systems, including SQLite and MySQL. SQLite is a lightweight, file-based database system suitable for smaller-scale applications, while MySQL is a robust, server-based database system capable of handling larger datasets and concurrent users. The form can be customized to support one or both of these database systems, providing flexibility and compatibility.

In summary, a student management form with add, update, delete, import, export, clear functions, and barcode generation capabilities provides a user-friendly interface to efficiently manage student information. It simplifies the process of recording and manipulating

student data, supports different database systems, and streamlines data import, export, and barcode generation operations.

Class management:-

A classes form is a graphical user interface (GUI) that allows users to manage classes information. This form typically includes fields for entering data such as the class ID, technology, class year, shift, and description. The form provides options for adding, updating, deleting, importing, exporting, and clearing data from the database.

The add function allows users to enter a new record into the database. In this case, users will enter information about a new class. The update function allows users to modify an existing record in the database. Users can change information such as the technology, class year, or shift for a specific class. The delete function allows users to remove a record from the database entirely.

The import function allows users to add data to the database in bulk by importing data from an external file, such as a spreadsheet or a CSV file. The export function allows users to save data from the database to an external file for use in other applications.

The clear function removes all data from the form and resets it to its default state. This can be useful when starting a new session or when needing to remove all data from the form.

The classes form can be designed to work with multiple database systems, such as SQLite and MySQL. SQLite is a lightweight, file-based database system that is often used for small-scale projects, while MySQL is a more robust, server-based database system that can handle larger

datasets and concurrent users. The form can be designed to support either or both of these database systems.

Overall, a classes form with add, update, delete, import, export, and clear functions can be a useful tool for managing class information in a user-friendly way. It can simplify data entry, retrieval, and modification, and allow users to work with multiple databases.

The screenshot shows a web application window titled 'Log Viewer'. On the left is a sidebar with a list of fields: ID, Technology, Class Name, Shift, and Details. Each field has a corresponding input box. The 'ID' box contains the value '1007'. Below the sidebar is a row of buttons: Add, Update, Delete, Clear, Import, and Export. The main area of the window displays a table with the following data:

ID	Technology	Class Name	Shift	Details
1003	Computer Information Techn	First Year	R1	First Year CIT R1
1004	Computer Information Techn	First Year	R2	First Year CIT R2
1005	Computer Information Techn	Second Year	R1	Second Year CIT R1
1006	Computer Information Techn	Second Year	R2	Second Year CIT R2
1001	Computer Information Techn	Third Year	R1	Third Year CIT R1
1002	Computer Information Techn	Third Year	R2	Third Year CIT R2

Felid:

- **Technology**
- **Class**
- **Shift**

Attendance management details

An attendance management form is a graphical user interface (GUI) tool that enables users to manage attendance data. This form typically consists of fields for capturing information such as the attendance ID, date, time, attendance status (in or out), student roll number, student name, technology, class year, and shift. The form provides functionalities for adding, updating, deleting, importing, exporting, and clearing data, with support for multiple databases such as SQLite and MySQL.

The add function allows users to input a new attendance record into the database. Users can enter details such as the date, time, attendance status, student roll number, student name, technology, class year, and shift for a particular attendance event. The update function enables users to modify an existing attendance record. Users can make changes to fields like date, time, attendance status, student details, or any other relevant information for a specific attendance entry.

The delete function permits users to remove an attendance record from the database entirely. This option can be used to remove any incorrect or unnecessary entries. The import function allows users to import attendance data in bulk from an external file, such as a CSV file. This feature streamlines the process of adding attendance information from various sources into the database. The export function allows users to save attendance data from the database to an external file. This exported data can be used for further analysis or reporting purposes.

The clear function is used to reset the attendance management form, removing all data and returning it to its default state. This functionality can be useful at the beginning of a new session or when starting fresh with a new set of attendance data.

The attendance management form is designed to be compatible with multiple database systems, including SQLite and MySQL. SQLite is a lightweight, file-based database system suitable for smaller-scale applications, while MySQL is a robust, server-based database system capable of handling larger datasets and concurrent users. The form can be tailored to support one or both of these database systems, providing flexibility and compatibility.

In summary, an attendance management form with add, update, delete, import, export, and clear functions provides a user-friendly interface to efficiently manage attendance data. It simplifies the process of recording and manipulating attendance information, supports different database systems, and streamlines data import and export operations.

Fees management

The screenshot displays a web application interface for managing fees. On the left, there is a 'Log Viewer' sidebar with a search bar and a list of fields: ID, Date, Time, Status, Student Roll Number, Student Name, Technology, Class, and Shift. The 'ID' field is currently set to '1003'. Below the sidebar is a row of buttons: Add, Update, Delete, Clear, Import, and Export. On the right, there is a table with the following columns: ID, Date, Time, Status, Student Roll Numb, Student Name, Technology, Class, and Shift. The table contains two rows of data:

ID	Date	Time	Status	Student Roll Numb	Student Name	Technology	Class	Shift
1001	03-04-2023	11:20	In	TCIT-R2-2020-419	Abdullah Ahmad	Computer Informat	TCIT	R2
1002	17-05-2023	8:00	In	TCIT-R2-2020-419	Abdullah Ahmad	Computer Informat	Third Year	R2

A fee management form is a graphical user interface (GUI) tool that enables users to manage fee-related information. This form typically includes fields such as fee ID, date, student roll number, student name, technology, class year, shift, total fee, and submitted fee. The form provides functionalities for adding, updating, deleting, importing from Excel CSV files, exporting to PDF, text, HTML, CSV, Excel XLSX formats, copying to the clipboard, and clearing data. It supports multiple databases such as SQLite and MySQL.

The add function allows users to input a new fee record into the database. Users can enter details such as the fee date, student roll number, student name, technology, class year, shift, total fee, and submitted fee. This function enables the addition of new fee records to track student payments.

The update function allows users to modify an existing fee record. Users can make changes to fields such as the fee date, student details, total fee, or submitted fee for

a specific fee record. This function ensures that the fee records are accurate and up to date.

The delete function allows users to remove a fee record from the database. This function is useful when a fee record becomes obsolete or when there are errors in the record.

The import function allows users to import fee data from an Excel CSV file. This feature simplifies the process of adding fee information in bulk from external sources into the database.

The export function enables users to export fee data in various formats, including PDF, text, HTML, CSV, and Excel XLSX. This functionality facilitates data sharing, reporting, and analysis by allowing users to save fee information in different formats that suit their specific needs.

The clear function resets the fee management form, removing all data and returning it to its default state. This function can be used to start fresh or when there is a need to clear all fee data from the form.

The fee management form supports multiple databases, including SQLite and MySQL. SQLite is a lightweight, file-based database system suitable for small-scale applications, while MySQL is a robust, server-based database system capable of handling larger datasets and concurrent users. The form can be customized to work with either or both of these database systems, providing flexibility and compatibility.

In summary, a fee management form with add, update, delete, import from Excel CSV, export to various formats, clear functions, and multiple database support simplifies the management of fee-related information. It provides an intuitive interface for tracking student payments, supports different databases, and offers

options for importing, exporting, and clearing fee data to meet diverse user requirements.

Settings

The settings for a student management system can vary depending on the specific requirements and functionalities. Here is a description of the settings you mentioned:

Advance Setting: This setting provides additional configuration options for advanced users. It may include features such as enabling/disabling specific modules, customizing system behavior, setting default values, configuring integration with other systems, and managing user roles and permissions.

Setting	Advance Setting
Remove Temp File After Close	True
Sqlite Database Backup	True
Sqlite Database Backup Every	Saturday
Sqlite Database Location	Default

Card Generator Setting (Student Card Generator): This setting specifically focuses on configuring the generation of student cards. It allows users to define the layout, design, and content of the student cards. Users can set parameters such as the card size, font styles, logo placement, student information to include, barcode integration, and other visual elements related to student identification.

Setting	Card Generator
Column Per Page	2
Row Per Page	4
Output In	PDF

Color Scheme: The color scheme setting allows users to customize the visual appearance of the student management system. Users can select or define color palettes for various elements of the user interface, such as backgrounds, buttons, text, headings, and other components. This setting enables personalization and branding of the system according to the user's preferences or organizational requirements.

Setting		Color Scheme
mainBackgroundColor	#282a36	
mainLightBackgroundColor	#44475a	
mainDarkBackgroundColor	#21222c	
mainForegroundColor	#f8f8f2	
mainButtonBackgroundColor	#6272a4	
mainButtonForegroundColor	#f8f8f2	
mainLabelBackgroundColor	#44475a	
mainLabelForegroundColor	#f8f8f2	
mainTextBackgroundColor	#44475a	
mainTextForegroundColor	#f8f8f2	
mainTextBorderColor	#6272a4	
mainTextHBackgroundColor	#6272a4	
mainComboBoxBackgroundColor	#44475a	
mainDTextBackgroundColor	#6272a4	
mainDTextForegroundColor	#f8f8f2	
mainDTextBorderColor	#f8f8f2	
mainTreeviewBackgroundColor	#44475a	
mainTreeviewForegroundColor	#f8f8f2	
mainTreeviewHBackgroundColor	#6272a4	
mainTreeviewHForegroundColor	#f8f8f2	
mainButtonActiveBackgroundColor	#bd93f9	
Save		Refresh
Clear		Default

Common Setting: Common settings refer to general configurations that apply to the entire student management system. It typically includes options like system language selection, date/time format, default currency, timezone settings, and other global preferences that affect the overall behavior and functionality of the system.

Setting	
Common Setting	
Messagebox After Add	False
After Add	Clear Record
Messagebox After Update	False
After Update	Clear Record
Messagebox After Delete	False
After Delete	Clear Record
Messagebox After Import	True
Messagebox After Export	True
Automatic Refresh After (ms)	1200
Refresh Thread	True
Full Screen	True
Color Scheme	Dracula Theme
Apply	

DB Configuration: The DB (database) configuration setting involves specifying the details required to establish a connection with the database system. Users can input parameters such as the database type (e.g., SQLite or MySQL), server address, port number, username, password, and database name. This setting ensures the student management system can interact with the chosen database and access or store data accurately.

Setting	
Database Configurations	
Database	sqlite
Host	None
User Name	None
Password	None
MySql Database Name	AS_DB
Test Database	Test

Form Setting: The form setting allows users to customize the appearance and behavior of the data entry forms within the student management system. Users can configure elements such as form layouts, field labels, field validations, data formatting options, and other form-related settings. This setting aims to adapt the forms to the specific needs and preferences of the user or organization.

Setting	Form Setting
Search Frame Height	50
Status Frame Height	20
Button and Detail Frame Width	410
Button Frame Height	30
Main Form Button Frame Width	300
Main Form Status Frame Height	20
Y Space In Fields	30
Width Of Fields	220
X Of Fields	170
ID Start With	1001
Main Form Height of Button	30
Main Heading Font	Algerian
Main Font	Calibri
Field Size	13
Field Height	25
Reset Status After 3s	True

Report Setting: The report setting focuses on customizing the generation and presentation of reports within the student management system. Users can define report templates, select data fields to include, specify sorting and filtering criteria, and set formatting options. This setting enables users to generate tailored reports, such as student attendance reports, fee payment summaries, academic performance reports, and more.

Setting	Report Setting
Report Save Automatically	True
Report Save Location	C:/Users/abdul/OneDrive/Desktop/Report
PDF Landscape Reports	<div> <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Delete"/> </div> <div> <!-- Empty list for Landscape Reports --> </div>
PDF Portrait Reports	<div> <input type="text"/> <input type="button" value="Add"/> <input type="button" value="Delete"/> </div> <div> <!-- Empty list for Portrait Reports --> </div>
PDF Report Size	A4
PDF Top Margin	25
PDF Bottom Margin	25
PDF Left Margin	25
PDF Right Margin	25
Open Report After Save	True

Pages Setting: The pages setting pertains to configuring the structure and content of the various pages or screens within the student management system. Users can define the navigation menu, create or edit pages, arrange page layouts, and manage access permissions for different user roles. This setting allows users to organize and customize the system's interface to suit their specific workflow and information requirements.

Overall, these settings provide administrators and users with flexibility and control over various aspects of the student management system, ensuring it can be tailored to meet specific needs, preferences, and organizational requirements.

Card generator

The image displays two screenshots of the Student Card Generator GUI. Both screenshots show a window titled 'Log Viewer' with a menu bar (File, More, Refresh) and a toolbar (Add, Generate, Delete, Clear, Import, Export).

Top Screenshot (Single Student Mode):

- The 'Generate' dropdown is set to 'Single Student'.
- Input fields are provided for Student Roll Number, Student Name, Technology, Class, Shift, and Code.
- The table titled 'STUDENT NAMES' is empty.

Bottom Screenshot (Multiple Student Mode):

- The 'Generate' dropdown is set to 'Multiple Student'.
- Input fields are provided for Technology, Class, and Shift.
- The table titled 'STUDENT NAMES' is empty.

The Student Card Generator form is a graphical user interface (GUI) tool that facilitates the creation of student cards based on different criteria

such as technology, class, shift, and individual students. The form provides functionalities for adding to the table for generating cards, deleting from the table, generating cards, clearing the table, importing data, and exporting generated cards. Here's a description of each functionality:

Add to Table for Generating: This function allows users to add students to a table for generating cards. Users can input details such as technology, class, shift, and other relevant student information into the form. The added students are listed in a table, ready for generating their respective cards.

Delete from Generating: This function enables users to remove selected students from the table for generating cards. Users can choose specific entries or use checkboxes to select multiple students and delete them from the list.

Generate: This function generates student cards based on the selected criteria. Users can choose from different options, such as generating cards for all students, generating cards based on technology and class, or generating cards based on technology, class, and shift. The generated cards can include student details, photographs, barcodes, and any other required information.

Clear: The clear function resets the Student Card Generator form, clearing the table and removing all data from the form. This allows users to start fresh or prepare the form for a new set of student card generation.

Import: The import function allows users to import student data from an external source, such as a CSV file. Users can import student details along with their respective technology, class, shift, and other relevant information. This feature simplifies the process of adding multiple students to the card generation table.

Export: The export function enables users to save the generated student cards to an external file or format. Users can export the cards as PDF, image files, or any other preferred format. This functionality facilitates printing or sharing the generated cards with other systems or applications.

The Student Card Generator form provides a user-friendly interface to efficiently manage and generate student cards based on various criteria. It offers flexibility in selecting students for card generation, supports importing and exporting data, and provides options to add, delete, generate, and clear student card data.

Scanner [attendance scanner page]

Our student management system incorporates a barcode scanning functionality that utilizes a webcam as the scanning device. This allows for the efficient and convenient scanning of student barcodes to record attendance in real-time. The process involves capturing an image using the webcam, processing it using image analysis techniques, detecting the student barcode within the image, and subsequently adding attendance records to the student management system's database.

Here is a breakdown of the steps involved:

Webcam Image Capture: The webcam captures an image that includes the area where the barcode is expected to be located. This image serves as the input for further processing.

Image Processing: The captured image undergoes image processing techniques to enhance its quality and facilitate barcode detection. This may involve operations such as resizing, noise reduction, contrast adjustment, and image filtering to optimize barcode readability.

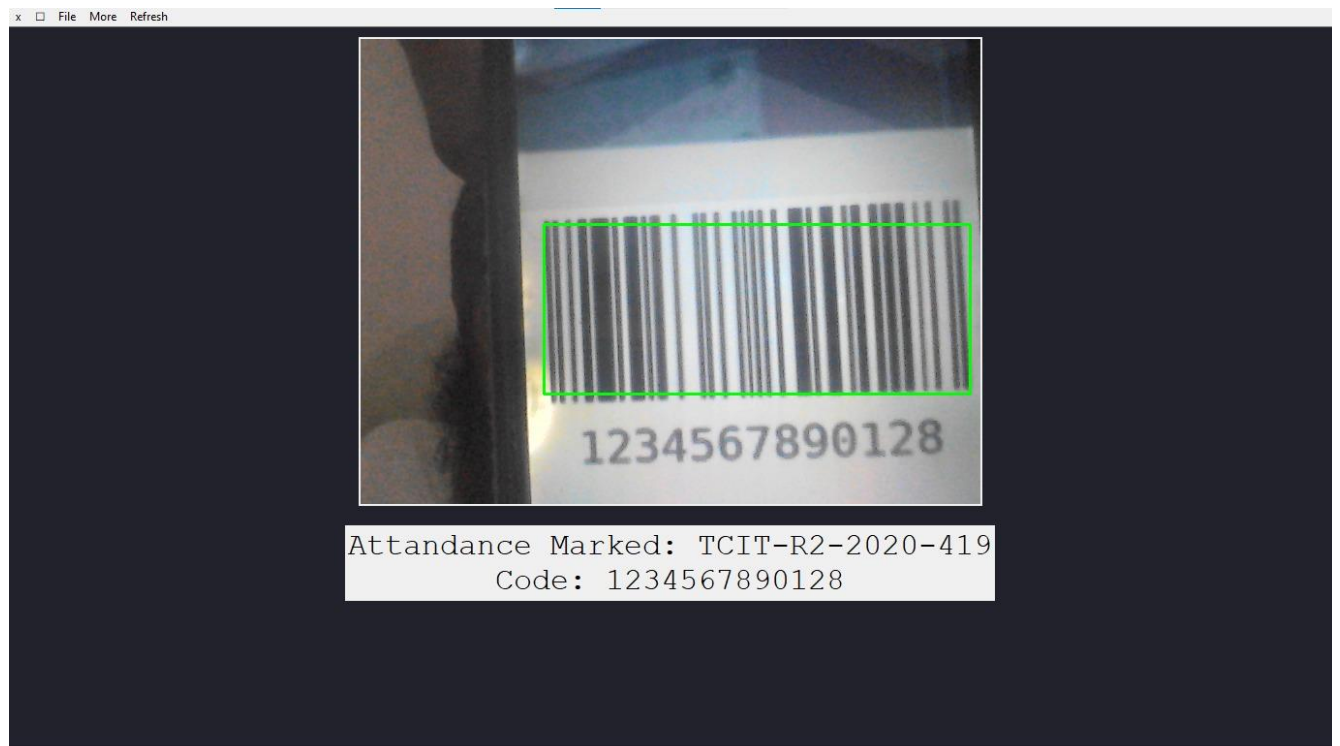
Barcode Detection: Image processing algorithms are employed to identify and locate the student barcode within the captured image. Techniques like edge detection, thresholding, and pattern recognition can be utilized to pinpoint the barcode's position accurately.

Barcode Decoding: Once the barcode is detected, a barcode decoding algorithm is applied to extract the encoded information from the barcode image. This algorithm interprets the pattern of bars and spaces, deciphering the barcode's unique identifier or data.

Attendance Recording: With the decoded barcode information, the student's attendance is recorded in real-time. The student management system's database is accessed, and an attendance record is added for the identified student. This process occurs instantly, providing immediate updates to the student's attendance status.

Database Integration: The attendance record is integrated into the student management system's database, ensuring that accurate attendance information is stored for future reference and analysis. The database can be updated with additional details, such as the date, time, and other relevant information related to the attendance event.

By utilizing a webcam as a barcode scanner and leveraging image processing techniques, this approach streamlines the attendance recording process. It eliminates the need for manual data entry, reducing errors and improving efficiency. The real-time nature of the system enables immediate attendance updates, providing up-to-date information for monitoring and analysis. This barcode scanning functionality enhances the overall functionality of the student management system and contributes to a streamlined and automated attendance management process.



Failed

There are problems, which we could not solve it.

First, the LCD not showing any command, which we give it.

Second, ftdi **driver is not connect with Esp32 cam.**

On other hands

When we attach Esp32 cam direct to Arduino.

Esp32 cam is working but without ftdi driver, we cannot connect LCD and Esp32 cam to Arduino.