Mini Project Report

On

"E-Returning System"

Submitted for the partial fulfillment of Bachelor of Engineering

By

1. **Kasula Thoshita 1SI17CS048**
2. **R P Manaswini Jain 1SI17CS079**
3. **Reethu Priya M S 1SI17CS087**

Under the guidance of

Mr. Prabodh C.P$_{\text{B.E, M.Tech.}}$

Assistant Professor, Dept of CSE

Department of Computer Science and Engineering

Siddaganga Institute of Technology, Tumakuru – 572103

(An Autonomous Institution, Affiliated to VTU, Belagavi & Recognized by AICTE, New Delhi)

2019 -2020

SIDDAGANGA INSTITUTE OF TECHNOLOGY,TUMAKURU-3

(An Autonomous Institution, Affiliated to VTU, Belagavi& Recognized by AICTE, New Delhi)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that that the mini project entitled **"E-Returning System"** is a bonafide work carried out by **Kasula Thoshita(1SI17CS048), R P Manaswini Jain(1SI17CS079), Reethu Priya M S(1SI17CS087)** of VI semester **Computer Science and Engineering**, **SIDDAGANGA INSTITUTE OF TECHNOLOGY** for the partial fulfillment of Bachelor of Engineering during the academic year 2019-2020.

Signature of the Guide
Mr. Prabodh C P
Assistant Professor

Signature of the Convener
Mr. Bhaskar G
Assistant Professor

Signature of the HOD
Dr. R Sumathi
Prof. and Head, Dept. of CSE

Name of the Examiners:Signature with Date

1.

2.

Acknowledgement

We consider this a privilege to express a few words of gratitude to all those who guided and inspired us for the successful completion of our project work.

With respectful prayers to His Holiness **Dr. Sree Sree Sree Shivakumara Swamigalu**, President, *Sree Siddaganga Education Society*, we express our gratitude at his lotus feet for being a constant source of inspiration in the course of study.

With gratitude we acknowledge our beloved *Directo*r **DR. M N Channabasappa** *M.A, Ph.D.*, and *Principal* **Dr.K.P.Shivananda** *M.E., Ph.D.*, for fostering an excellent academic environment in this institution and also providing excellent lab facilities, which made our endeavor possible.

We express our deep sense of gratitude to **Dr. R. Sumathi** *M.E., Ph.D., Professor and Head, Department of Computer Science and engineering*, who has always been a remarkable source of inspiration whose unflagging zeal has catalyzed us to strive for greater heights during the span of course in this Institution.

We thank our project guide **Mr. Prabodh C P** *M.Tech, Assistant Professor, Department of Computer Science and Engineering* for providing the critical support on every step of this project development. His guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work. Without his aspiring support this project would not have been implemented.

We thank our project coordinators for giving us a platform to showcase the different skills that we have in order to complete the mini project.

Finally, we would like to thank all the faculty and non-teaching staff members of *Dept. of Computer Science and Engineering* and our parents and friends, without whose help and constant encouragement, it would have been difficult for us to complete this project.

ABSTRACT

E – Returning system is an integrated system, designed to impart library book returning system in an effective and hustle free way. It helps students to return their books faster without the supervision of librarian and also is an alternate for manual fine collection. This integrated system consists of display, barcode reader, money storage vault and book storage vault. The application software is developed in such a way that it takes restricted input from the user and gets them through the screen.

Basically, this application takes book code as the input and matches it with the pre-stored details in the database and displays the user incase of fine else accepts the book and disposes into book storage vault. Our one of the focuses is to reduce the manual efforts and also the chaos created in providing the exact change during fine collection. So, we have included hustle free change-exchange feature. When user has to pay the fine, he/she can deposit it at the deposit port and then system carries out image processing for detecting the cash say coins/notes. After determining the value of coins/notes deposited respective validations will be performed.

Table of Contents

CHAPTER 1

INTRODUCTION

The Project titled “E-Returning System” is an Application Software for monitoring and controlling the transactions in library. This Project is developed using Flask, and mainly focuses on user friendly interface and quick automation of existing returning system. Our Software is easy to use for both beginners and advanced users. It features a familiar and well-thought-out, an attractive user interface, combined with strong searching and reporting capabilities.

The Project “E-Returning System’’ gives us information how the library system works. Firstly, Book ID has to be entered and it is matched with database. If the match occurs the system will return the details of the Student like USN, date of issue, due date and fine amount. Here, the Student will be given an option to either proceed further to pay or to leave.There will be a tutorial video beside the book details on how to input the money into the system. While paying the fine the money is deposited in the provided vault, here the image of money is taken and image processing is carried out. If it is sufficient for the fine it will accept and payment will complete. It there is no penalty then the book will be accepted.

1.1 Motivation

Sophisticated Technologies motivates the thought of having an advanced model in the college. Basically, this Project is for the students not to waste their time by waiting in queue, because we have enough experience in our college while returning a book, so to overcome the difficulties we have designed this e-return system.The model moderates the waiting time in the library counter during end of the semester. It decreases the fuss that is created for the demand of exact change. This is best suited for libraries in schools and colleges.

1.2 Aim of the Project

The main aim of this project is providing an easy to handle and automated library management system. Manual efforts involved in current library system are high and the rate of work to be done increases during the last working days of the semester. In addition, if library account lacks physical money then the delay will be increased so, the system doesn’t seem to be efficient. Hence replacing the system with the prescribed model can take care of the issues.

CHAPTER 2

LITERATURE SURVEY

Software is a collection of data or computer instructions that tell the computer how to work. There are two basic type camps of software development: **Application Development and System Development**. Application Development is focused on creating programs that meet the user's needs. Our project comes under Application software.

2.1 FIREBASE

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it.

Firebase's first product was the Firebase Real-time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

The database is also accessible through a REST API and bindings for several JavaScript frameworks, such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-SentEvents protocol, which is an API creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company's server-side enforced security rules.

We are making use of firebase to store the details of Book id, student id for whom the book is issued, issued date, due date.

2.2 FLASK – Web Framework

Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004.Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation or any other components where pre-existing third-party libraries provide common functions.

Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, and upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

Generally, frameworks provide support for a number of activities such as interpreting requests producing responses storing data persistently, and so on. As a developer using a framework, one typically write code which conforms to some kind of conventions that lets you "plug in" to the framework, delegating responsibility for the communications, infrastructure and low-level stuff to the framework while concentrating on the logic of the application in your own code. Many frameworks now provide an element of customization in their support for the above activities and abstractions, utilizing components in that they provide abstractions only for certain specific things. As a result, it can be possible for you to build your own full-stack framework almost entirely from existing components

2.3 JINJA2

Jinja is a web template engine for the Python programming language. It is licensed under a BSD License. Jinja is similar to the Django template engine but provide Python-like expressions while ensuring that the templates are evaluated in a sandbox. It is a text-based template language and thus can be used to generate any markup as well as source code.

The Jinja template engine allows customization of tags, filters, tests, and global. Also, unlike the Django template engine, Jinja allows the template designer to call functions with arguments on objects Jinja is Flask's default template engine.

Some features of Jinja are:

1. Sandboxed execution

2. Automatic HTML escaping to prevent cross-site scripting (XSS) attacks.

3. Template inheritance.

4. Compiles down to the optimal Python code just-in-time.

5. Optional ahead-of-time template compilation.

6. Easy to debug.

7. Configurable syntax.

Jinja’s philosophy is that while application logic belongs in Python if possible, it shouldn’t make the template designer’s job difficult by restricting functionality too much.

2.4 ARDUINO UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "Uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

2.5 PYDUINO

Python library to interface with arduino via serial connection.

Functionality Implemented:

The library implements a two-way communication over the serial connection with the Arduino, sending text strings that encode operations to be performed in the Arduino board and parsing the returned messages.

The functionalities of the Arduino library that are currently exposed via the API are:

- pinMode()
- digitalRead()
- digitalWrite()
- analogRead()
- analogWrite()

2.6 IMAGE PROCESSING

Image processing is a method to perform some operations on an image. It is a type of signal processing in which input is an image and output may be image or features associated with that image.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools;
- Analyzing and manipulating the image;
- Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

2.7 TENSORFLOW

TensorFlow is a Python-friendly open source library for numerical computation that makes machine learning faster and easier. Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor. TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together.

The single biggest benefit TensorFlow provides for machine learning development is abstraction. Instead of dealing with the nitty-gritty details of implementing algorithms, or figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application. TensorFlow takes care of the details behind the scenes. TensorFlow offers additional conveniences for developers who need to debug and gain introspection into TensorFlow apps.

2.8 YOLO V3 Architecture

YOLO v3 uses a variant of Darknet, which originally has 53 layer network trained on Imagenet. For the task of detection, 53 more layers are stacked onto it, giving us a 106 layer fully convolutional underlying **architecture** for **YOLO v3.**

It is one of the faster object detection algorithm, though it is no longer the most accurate object detection algorithm, it is a very good choice when you need real-time detection, without loss of too much accuracy. **The most salient feature of v3 is that it makes detections at three different scales.** In YOLO v3, **the detections done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network.** This Algorithm applies a single neural network to the Full Image. It means that this network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

2.9 IMAGEAI

ImageAI is a python library built to empower developers and students to build applications and systems with self-contained Deep Learning and Computer Vision capabilities using simple and few lines of code. ImageAI is developed by Moses Olafenwa, John Olafenwa and Deep Quest AI team. It provides very powerful and easy to use classes to perform Image Recognition tasks. We can perform all of these state-of-the-art computer vision tasks with python code. Once we have Python, other dependencies and ImageAI installed on our computer system, there is no limit to the incredible applications. There is some classes can be integrated into any traditional python program you are developing, be it a website, Windows/Linux/MacOS application.

ImageAI allows performing all of these with state-of-the-art deep learning algorithms like RetinaNet, YOLOv3 and TinyYOLOv3, With ImageAI we can run detection tasks and analyze images. It also provides very powerful yet easy to use classes and functions to perform Video Object Detection and Tracking and Video analysis.

2.10 LABELIMG

Labelimg is a free, open source tool for graphically labeling images. It is written in Python and uses QT for its graphical interface. It is an easy, free way to label a few hundred images. It supports labelling in VOC XML or YOLO text file format. At Roboflow, we strongly recommend you use the default VOC XML format for creating labels. VOC XML is a more universal standard as it relates to object detection whereas various YOLO implementations have slightly different text file formats. Moreover, you can always easily convert from VOC XML to any other format using Roboflow, like vocoxml to coco json.

CHAPTER 3

REQUIREMENT SPECIFICATION

In Software engineering, application developers need to clearly understand the problems to be solved. It is therefore important for a developer to properly model the scenarios that can influence the solution to the problem by collecting relevant information. This process is called requirement analysis. The requirement analysis provides the opportunity for a developer to get a better understanding of the problem in question. For effective design and development of this project, the following requirements must be met. They can be divided into functional requirements and non-functional requirements.

3.1 Functional Requirements:

This section describes different requirements that are accomplished E-Returning System. In order to achieve the desired goals of this project, the functional requirements must be met. The following are the major actions performed by E-Returning System.

Input Requirements:

Input requirements are the requirements that a user must fulfill to access the E-Returning System. As this application is built for the use of students to support them in returning the books issued to them by library, students are requested to scan the barcode on the book using the port were barcode reader is placed. Also, in case of any payment to be made as a part of fine they are supposed to input the money (coins/notes) in the vault specified for money collection. Where an image of the deposited money will be taken and further processed to get the value deposited. Later on validations like value greater than the fine value then the rest change has to be returned back and incase if the value is less than the fine value rejecting both the book and money inserted alerting insufficient money for the request.

3.2 Non-Functional Requirements:

Non-Functional requirements are the requirements that do not affect the proper running of E-Returning System. However, it is worthwhile to mention and consider these requirements for the purpose of software quality and analysis.

Software Requirements:

E-Returning System like other software engineering projects needs well defined application that must meet the software environment needed to achieve the desired goal of the project. The software requirements considered in the development of this project are highlighted below:

1. For this web application we need an operating system: Windows Server.
2. Image Processing Technique: Tensorflow.

External Interface or Hardware Requirements:

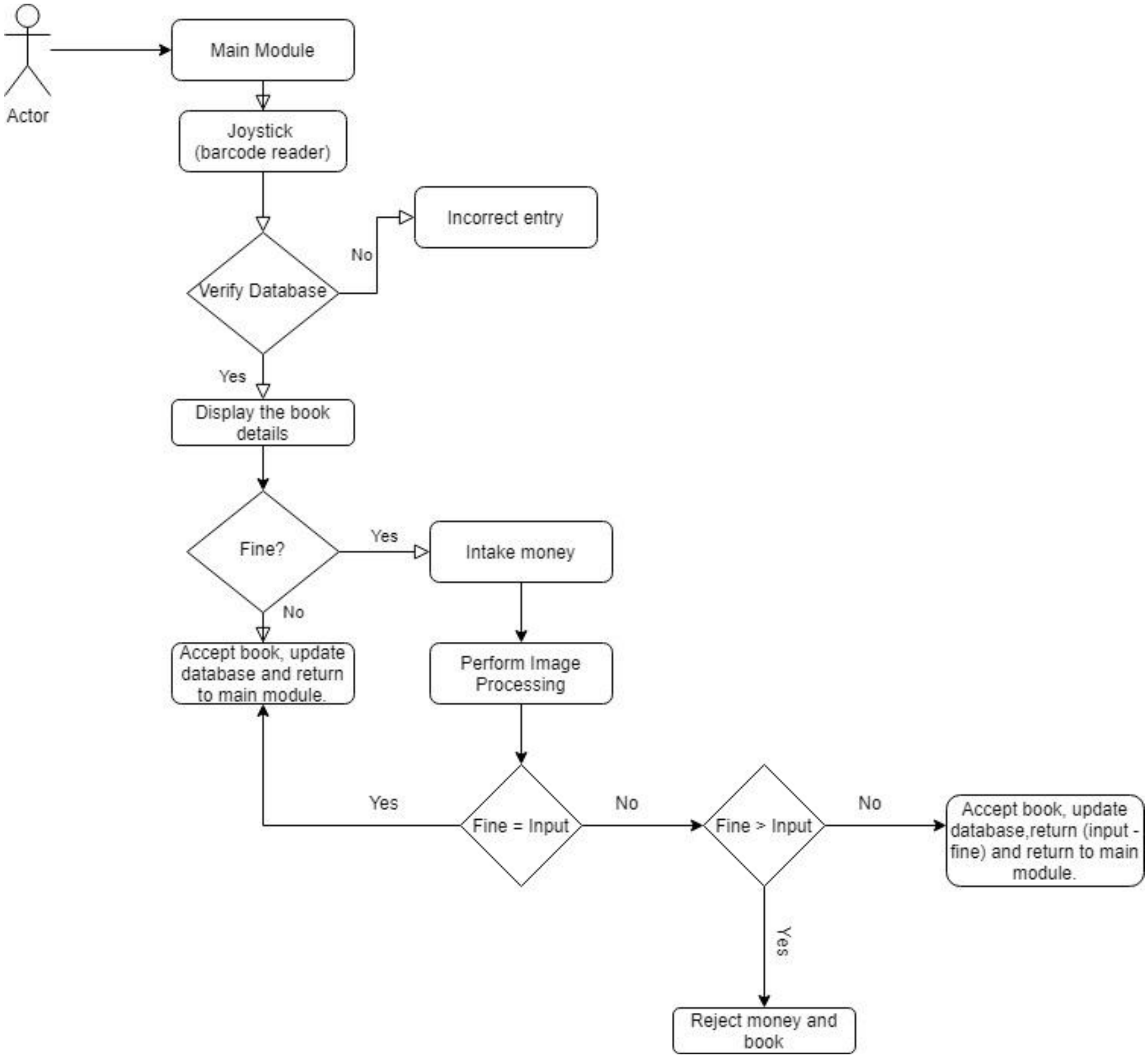
In the development of this application certain hardware requirements and specifications were considered for the application to be functional and result oriented. This application utilizes a camera for capturing images of money deposited, a joystick for scanning the barcode over the book, and arduino along with some led’s for indicating the vaults of the machine, sensors for opening and closing the vaults when needed.

CHAPTER 4

DESIGN & IMPLEMENTATION

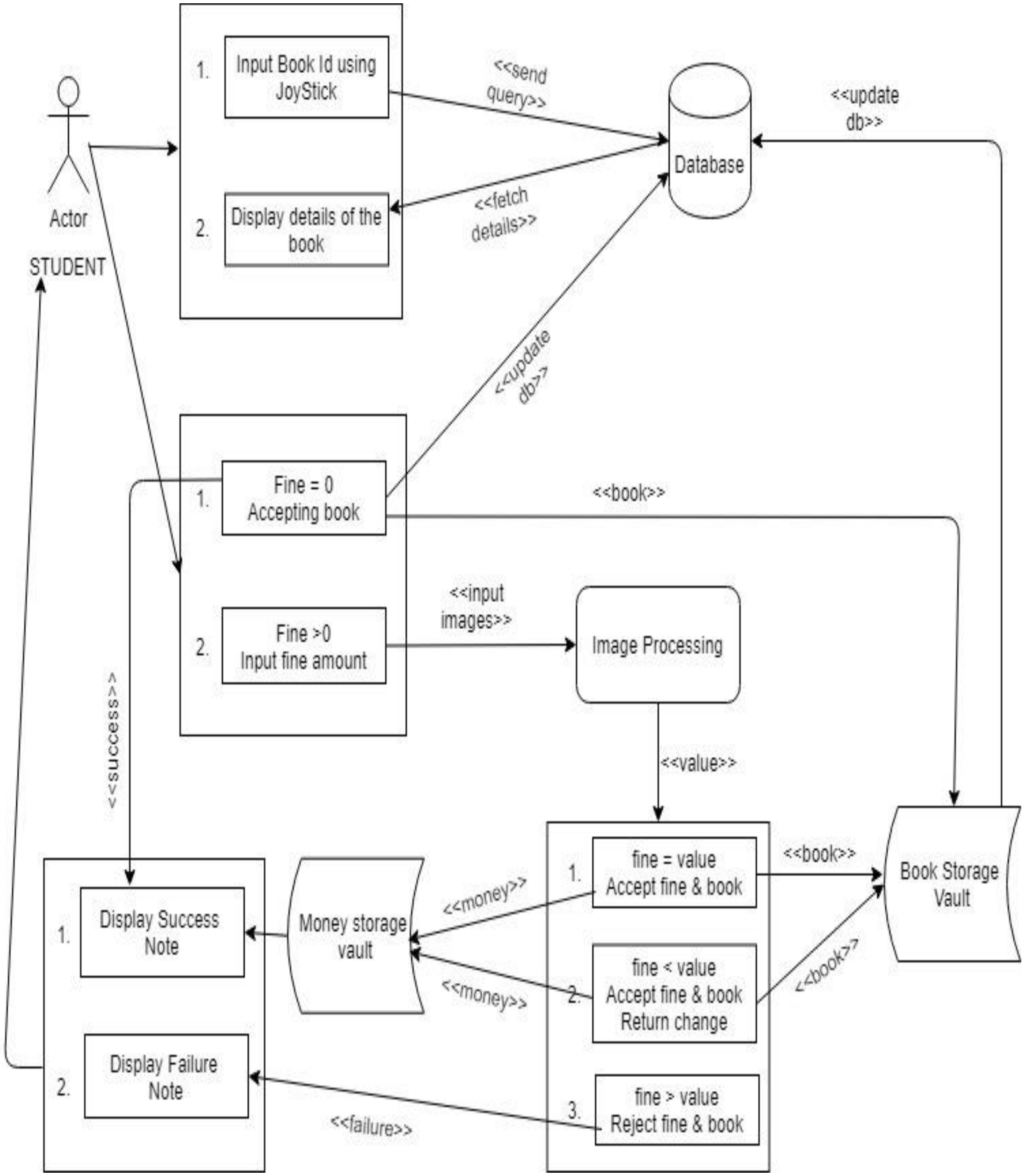
4.1 DESIGN:

FLOW CHART:

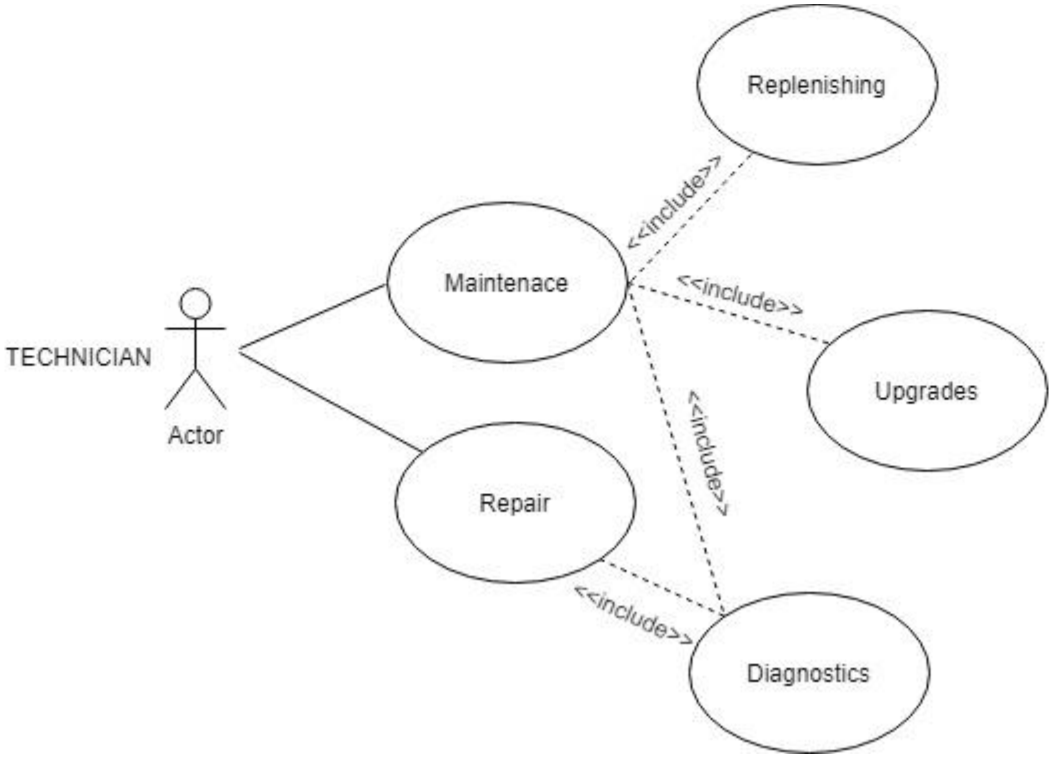


USE CASE DIAGRAM:

For Student accessing the E-Returning System –



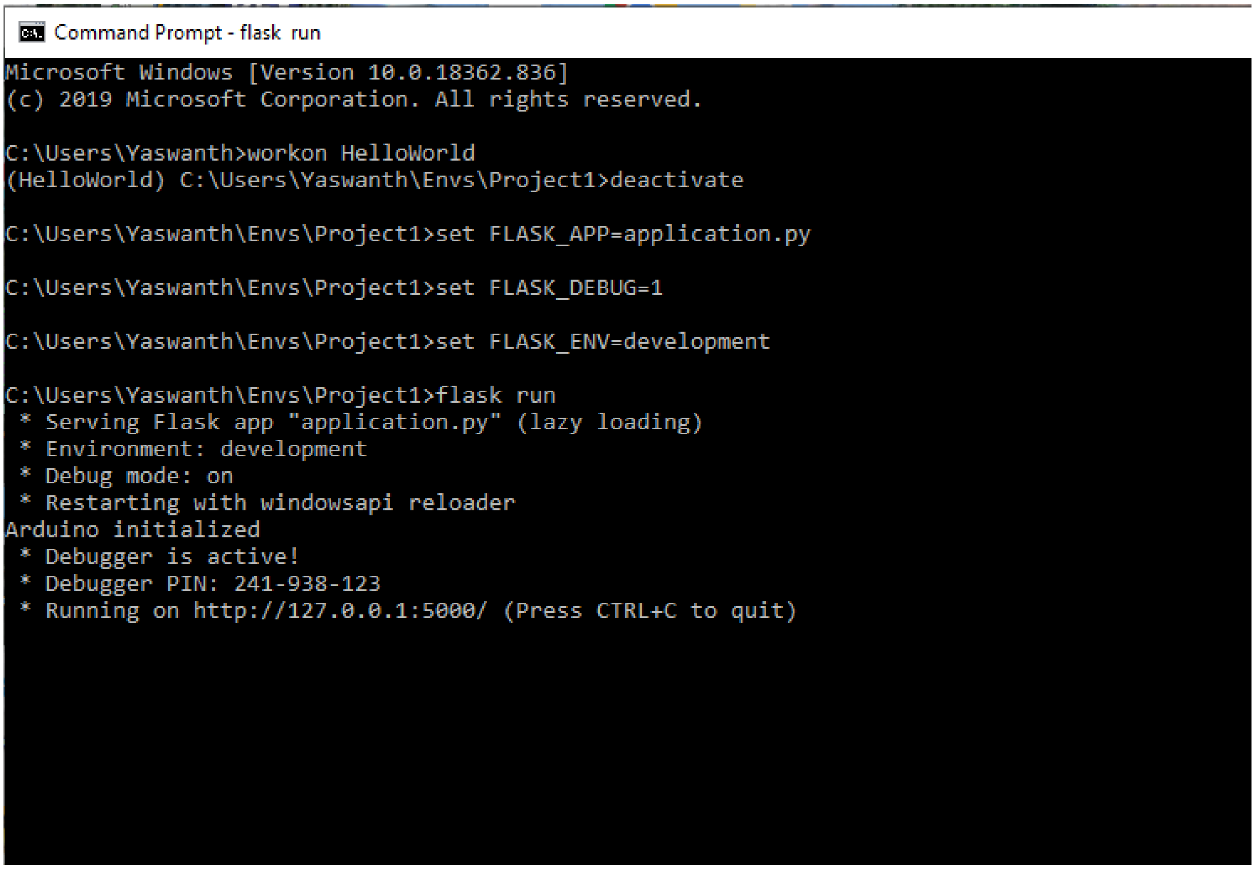
For Technical Maintenance of the E-Returning System –



4.2 IMPLEMENTATION OF MODULES:

(a) Initializing system –

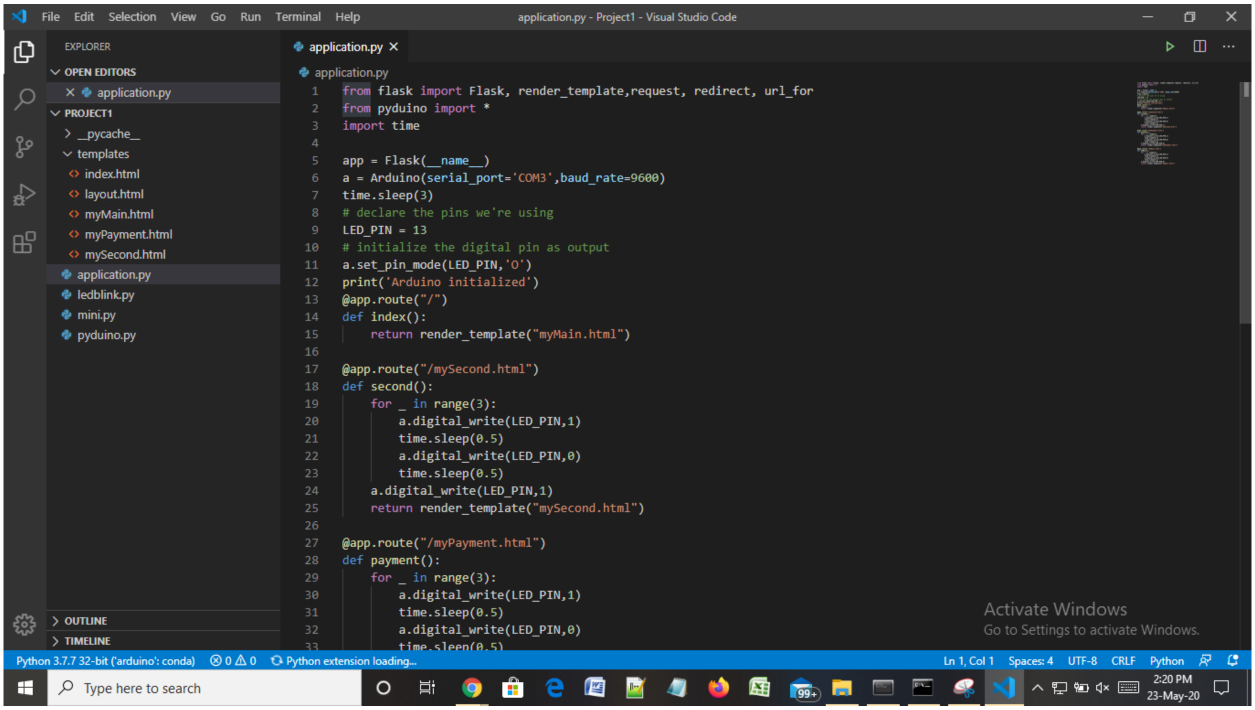
In this phase we are setting up flask to run our application.



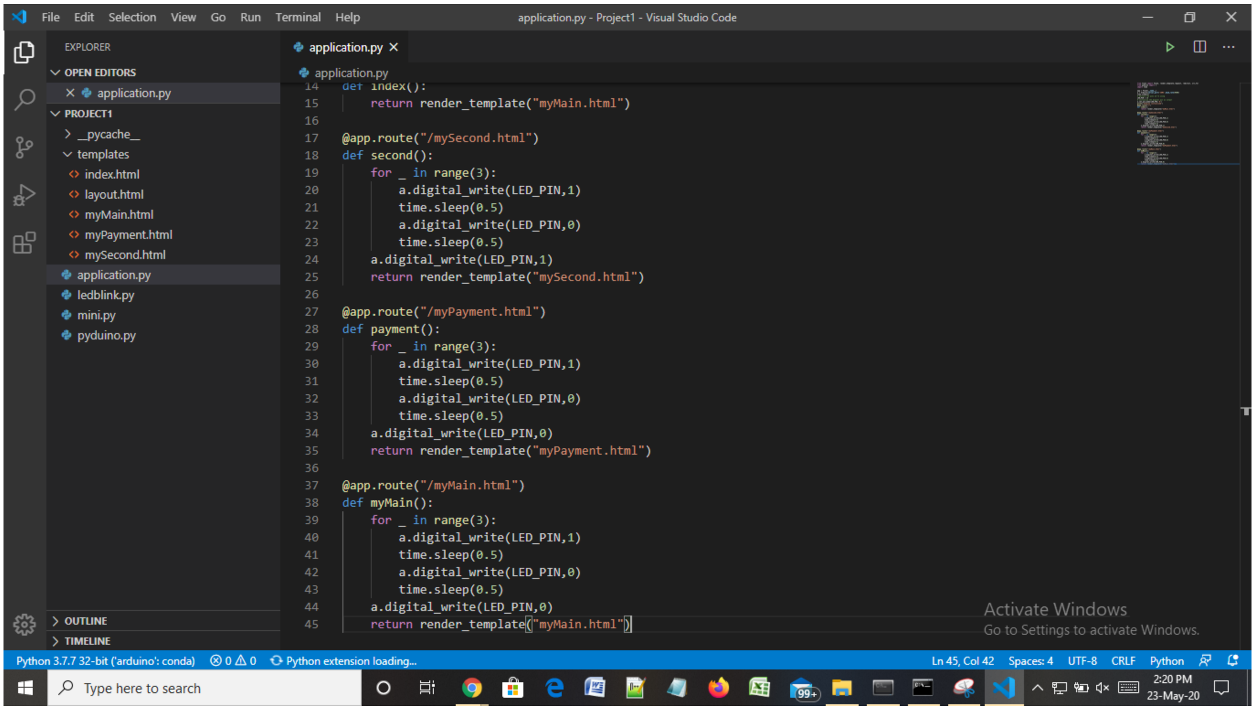
Command Prompt - flask run

```
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Yaswanth>workon HelloWorld
(HelloWorld) C:\Users\Yaswanth\Envs\Project1>deactivate

C:\Users\Yaswanth\Envs\Project1>set FLASK_APP=application.py

C:\Users\Yaswanth\Envs\Project1>set FLASK_DEBUG=1

C:\Users\Yaswanth\Envs\Project1>set FLASK_ENV=development

C:\Users\Yaswanth\Envs\Project1>flask run
 * Serving Flask app "application.py" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Restarting with windowsapi reloader
Arduino initialized
 * Debugger is active!
 * Debugger PIN: 241-938-123
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

(b) application.py module –

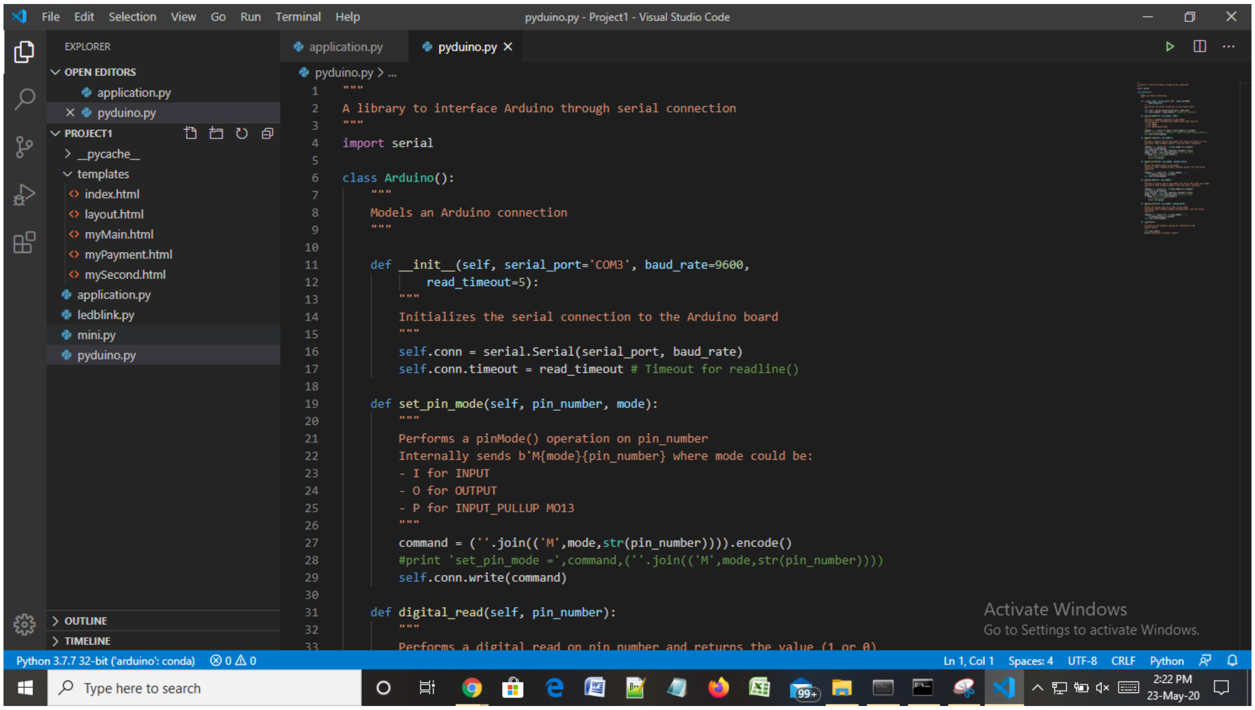

File Edit Selection View Go Run Terminal Help
application.py - Project1 - Visual Studio Code
EXPLORER
OPEN EDITORS
application.py
PROJECT1
pycache
templates
index.html
layout.html
myMain.html
myPayment.html
mySecond.html
application.py
ledblink.py
mini.py
pyduino.py
OUTLINE
TIMELINE
Activate Windows
Go to Settings to activate Windows.
Python 3.7.7 32-bit ('arduino': conda)
Python extension loading...
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
2:20 PM
23-May-20

```python
from flask import Flask, render_template,request, redirect, url_for
from pyduino import *
import time

app = Flask(__name__)
a = Arduino(serial_port='COM3',baud_rate=9600)
time.sleep(3)
# declare the pins we're using
LED_PIN = 13
# initialize the digital pin as output
a.set_pin_mode(LED_PIN,'O')
print('Arduino initialized')
@app.route("/")
def index():
    return render_template("myMain.html")

@app.route("/mySecond.html")
def second():
    for _ in range(3):
        a.digital_write(LED_PIN,1)
        time.sleep(0.5)
        a.digital_write(LED_PIN,0)
        time.sleep(0.5)
    a.digital_write(LED_PIN,1)
    return render_template("mySecond.html")

@app.route("/myPayment.html")
def payment():
    for _ in range(3):
        a.digital_write(LED_PIN,1)
        time.sleep(0.5)
        a.digital_write(LED_PIN,0)
        time.sleep(0.5)
```

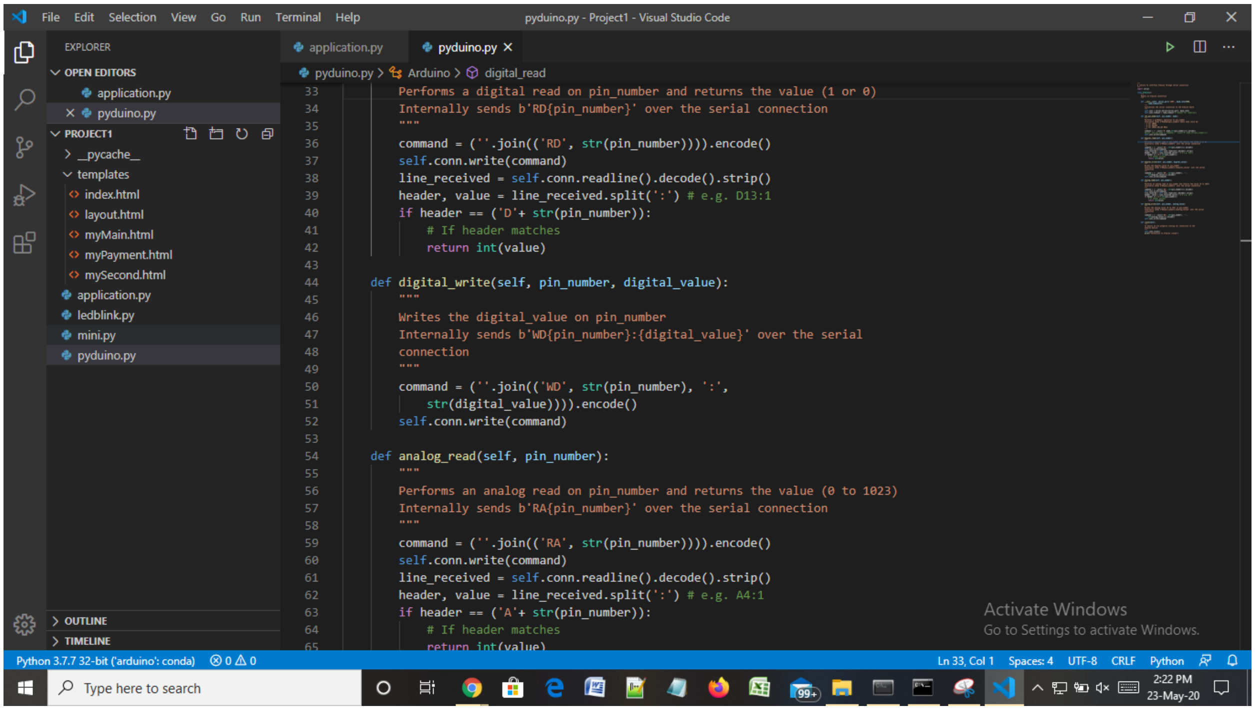

File Edit Selection View Go Run Terminal Help
application.py - Project1 - Visual Studio Code
EXPLORER
OPEN EDITORS
application.py
PROJECT1
pycache
templates
index.html
layout.html
myMain.html
myPayment.html
mySecond.html
application.py
ledblink.py
mini.py
pyduino.py
OUTLINE
TIMELINE
Activate Windows
Go to Settings to activate Windows.
Python 3.7.7 32-bit ('arduino': conda)
Python extension loading...
Ln 45, Col 42 Spaces: 4 UTF-8 CRLF Python
Type here to search
2:20 PM
23-May-20

```python
def index():
    return render_template("myMain.html")

@app.route("/mySecond.html")
def second():
    for _ in range(3):
        a.digital_write(LED_PIN,1)
        time.sleep(0.5)
        a.digital_write(LED_PIN,0)
        time.sleep(0.5)
    a.digital_write(LED_PIN,1)
    return render_template("mySecond.html")

@app.route("/myPayment.html")
def payment():
    for _ in range(3):
        a.digital_write(LED_PIN,1)
        time.sleep(0.5)
        a.digital_write(LED_PIN,0)
        time.sleep(0.5)
    a.digital_write(LED_PIN,0)
    return render_template("myPayment.html")

@app.route("/myMain.html")
def myMain():
    for _ in range(3):
        a.digital_write(LED_PIN,1)
        time.sleep(0.5)
        a.digital_write(LED_PIN,0)
        time.sleep(0.5)
    a.digital_write(LED_PIN,0)
    return render_template("myMain.html")
```

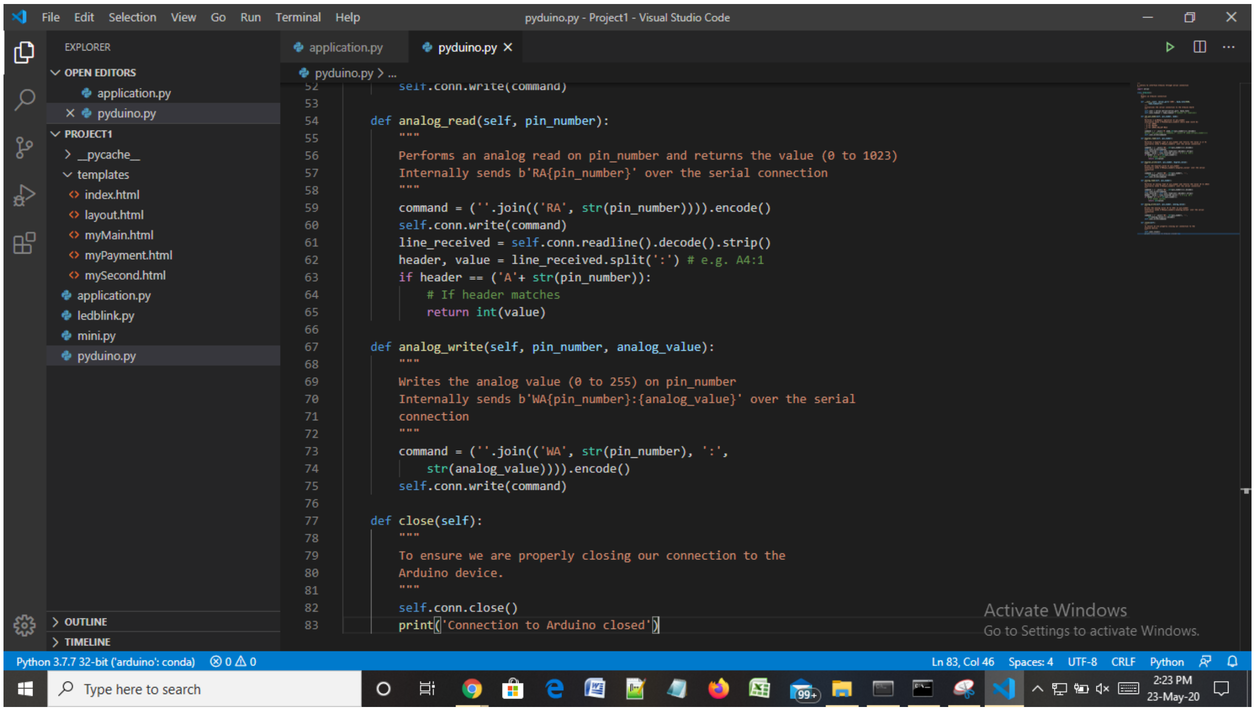
(c) pyduino.py module –



```
"""
A library to interface Arduino through serial connection
"""
import serial

class Arduino():
    """
    Models an Arduino connection
    """

    def __init__(self, serial_port='COM3', baud_rate=9600,
            read_timeout=5):
        """
        Initializes the serial connection to the Arduino board
        """
        self.conn = serial.Serial(serial_port, baud_rate)
        self.conn.timeout = read_timeout # Timeout for readline()

    def set_pin_mode(self, pin_number, mode):
        """
        Performs a pinMode() operation on pin_number
        Internally sends b'M{mode}{pin_number} where mode could be:
        - I for INPUT
        - O for OUTPUT
        - P for INPUT_PULLUP MO13
        """
        command = (''.join(('M',mode,str(pin_number)))).encode()
        #print 'set_pin_mode =',command,(''.join(('M',mode,str(pin_number))))
        self.conn.write(command)

    def digital_read(self, pin_number):
        """
        Performs a digital read on pin_number and returns the value (1 or 0)
```



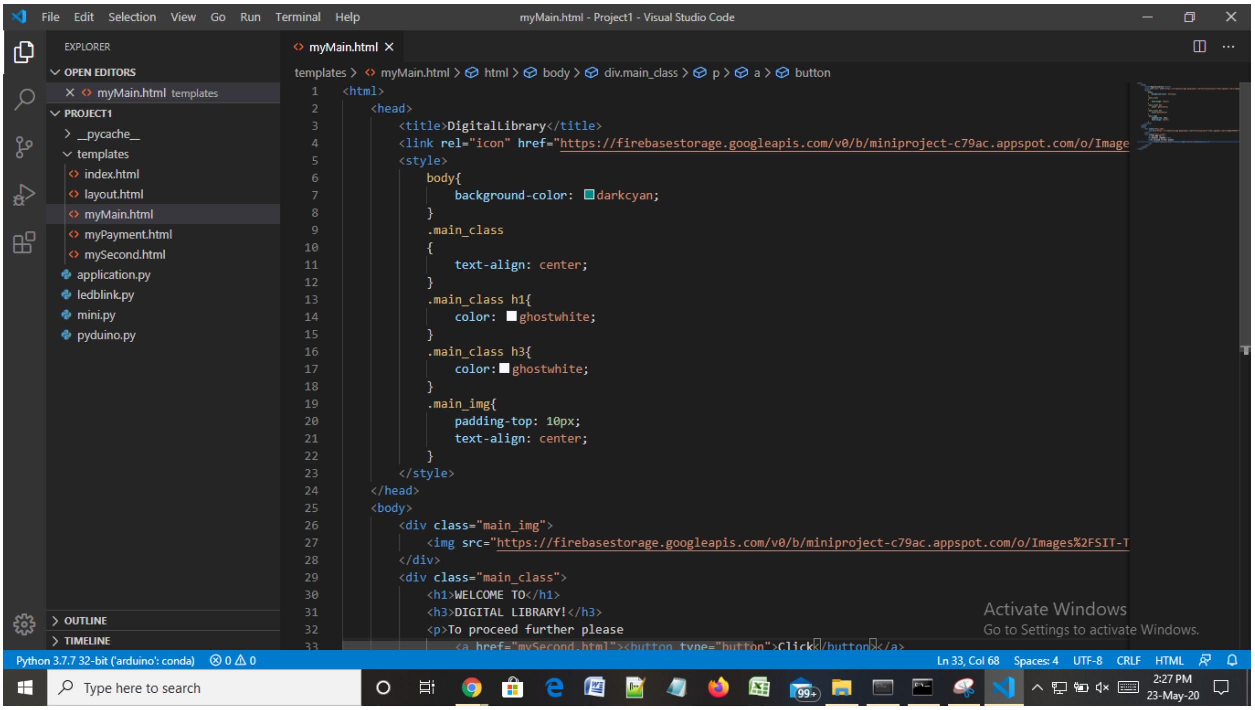
```
        Performs a digital read on pin_number and returns the value (1 or 0)
        Internally sends b'RD{pin_number}' over the serial connection
        """
        command = (''.join(('RD', str(pin_number)))).encode()
        self.conn.write(command)
        line_received = self.conn.readline().decode().strip()
        header, value = line_received.split(':') # e.g. D13:1
        if header == ('D'+ str(pin_number)):
            # If header matches
            return int(value)

    def digital_write(self, pin_number, digital_value):
        """
        Writes the digital_value on pin_number
        Internally sends b'WD{pin_number}:{digital_value}' over the serial
        connection
        """
        command = (''.join(('WD', str(pin_number), ':',
            str(digital_value)))).encode()
        self.conn.write(command)

    def analog_read(self, pin_number):
        """
        Performs an analog read on pin_number and returns the value (0 to 1023)
        Internally sends b'RA{pin_number}' over the serial connection
        """
        command = (''.join(('RA', str(pin_number)))).encode()
        self.conn.write(command)
        line_received = self.conn.readline().decode().strip()
        header, value = line_received.split(':') # e.g. A4:1
        if header == ('A'+ str(pin_number)):
            # If header matches
            return int(value)
```



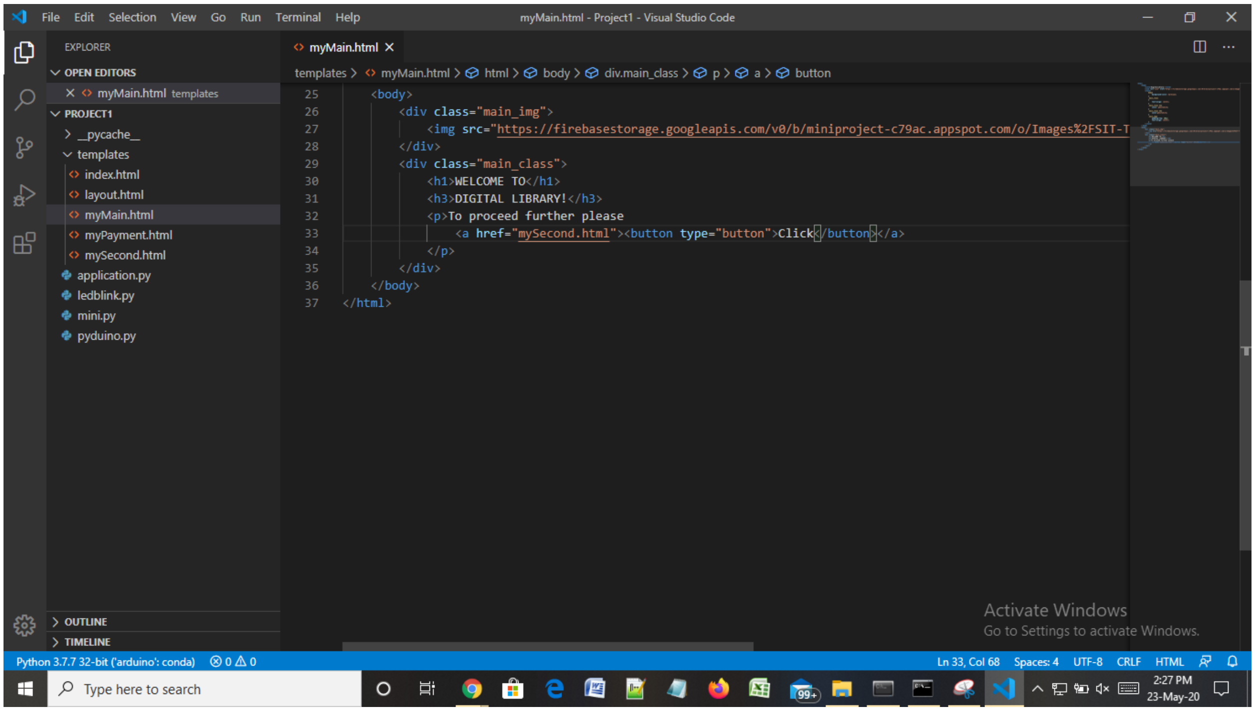
```python
            self.conn.write(command)

    def analog_read(self, pin_number):
        """
        Performs an analog read on pin_number and returns the value (0 to 1023)
        Internally sends b'RA{pin_number}' over the serial connection
        """
        command = (''.join(('RA', str(pin_number)))).encode()
        self.conn.write(command)
        line_received = self.conn.readline().decode().strip()
        header, value = line_received.split(':') # e.g. A4:1
        if header == ('A'+ str(pin_number)):
            # If header matches
            return int(value)

    def analog_write(self, pin_number, analog_value):
        """
        Writes the analog value (0 to 255) on pin_number
        Internally sends b'WA{pin_number}:{analog_value}' over the serial
        connection
        """
        command = (''.join(('WA', str(pin_number), ':',
            str(analog_value)))).encode()
        self.conn.write(command)

    def close(self):
        """
        To ensure we are properly closing our connection to the
        Arduino device.
        """
        self.conn.close()
        print('Connection to Arduino closed')
```

(d) Main module –

In this module we just provide user to click to continue for experiencing the automatic book return system.


File Edit Selection View Go Run Terminal Help
myMain.html - Project1 - Visual Studio Code
EXPLORER
OPEN EDITORS
myMain.html templates
PROJECT1
pycache
templates
index.html
layout.html
myMain.html
myPayment.html
mySecond.html
application.py
ledblink.py
mini.py
pyduino.py
OUTLINE
TIMELINE
myMain.html
templates > myMain.html > html > body > div.main_class > p > a > button
Activate Windows
Go to Settings to activate Windows.
Python 3.7.7 32-bit ('arduino': conda) ⊗ 0 ⚠ 0
Ln 33, Col 68 Spaces: 4 UTF-8 CRLF HTML
Type here to search
2:27 PM
23-May-20

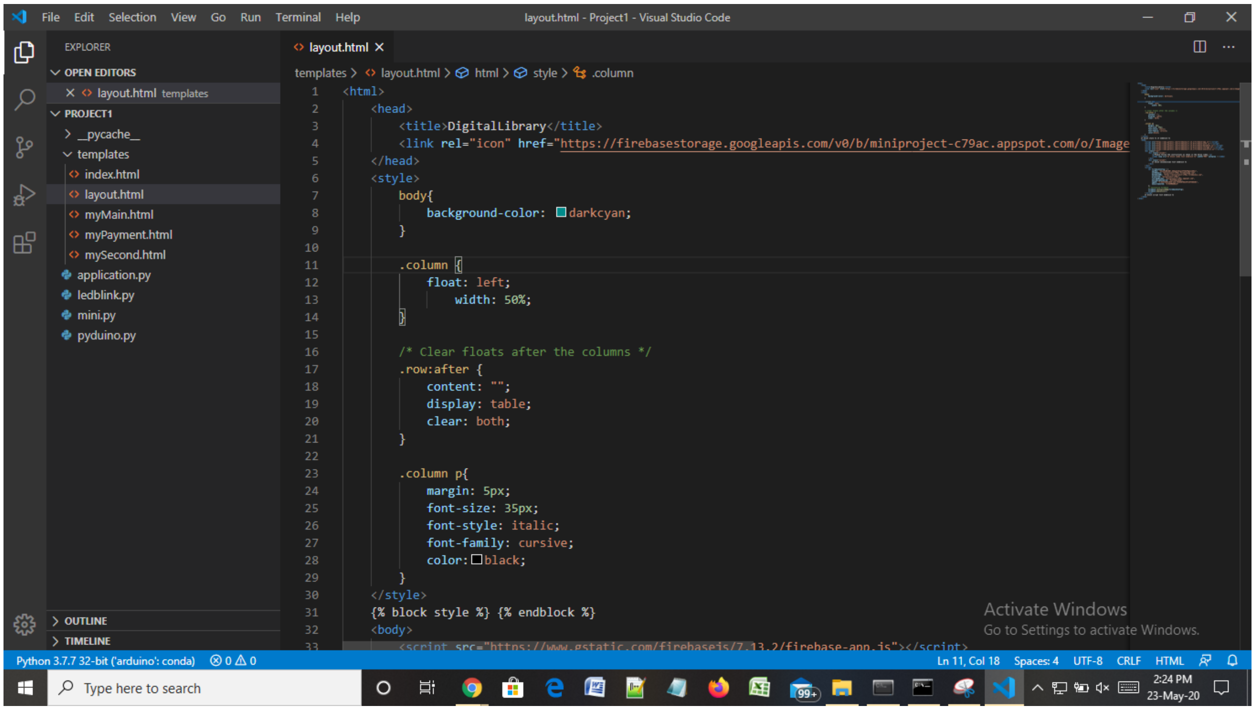
```html
<html>
    <head>
        <title>DigitalLibrary</title>
        <link rel="icon" href="https://firebasestorage.googleapis.com/v0/b/miniproject-c79ac.appspot.com/o/Image
        <style>
            body{
                background-color: darkcyan;
            }
            .main_class
            {
                text-align: center;
            }
            .main_class h1{
                color: ghostwhite;
            }
            .main_class h3{
                color: ghostwhite;
            }
            .main_img{
                padding-top: 10px;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <div class="main_img">
            <img src="https://firebasestorage.googleapis.com/v0/b/miniproject-c79ac.appspot.com/o/Images%2FSIT-T
        </div>
        <div class="main_class">
            <h1>WELCOME TO</h1>
            <h3>DIGITAL LIBRARY!</h3>
            <p>To proceed further please
                <a href="mySecond.html"><button type="button">Click</button></a>
```


File Edit Selection View Go Run Terminal Help
myMain.html - Project1 - Visual Studio Code
EXPLORER
OPEN EDITORS
myMain.html templates
PROJECT1
pycache
templates
index.html
layout.html
myMain.html
myPayment.html
mySecond.html
application.py
ledblink.py
mini.py
pyduino.py
OUTLINE
TIMELINE
myMain.html
templates > myMain.html > html > body > div.main_class > p > a > button
Activate Windows
Go to Settings to activate Windows.
Python 3.7.7 32-bit ('arduino': conda) ⊗ 0 ⚠ 0
Ln 33, Col 68 Spaces: 4 UTF-8 CRLF HTML
Type here to search
2:27 PM
23-May-20

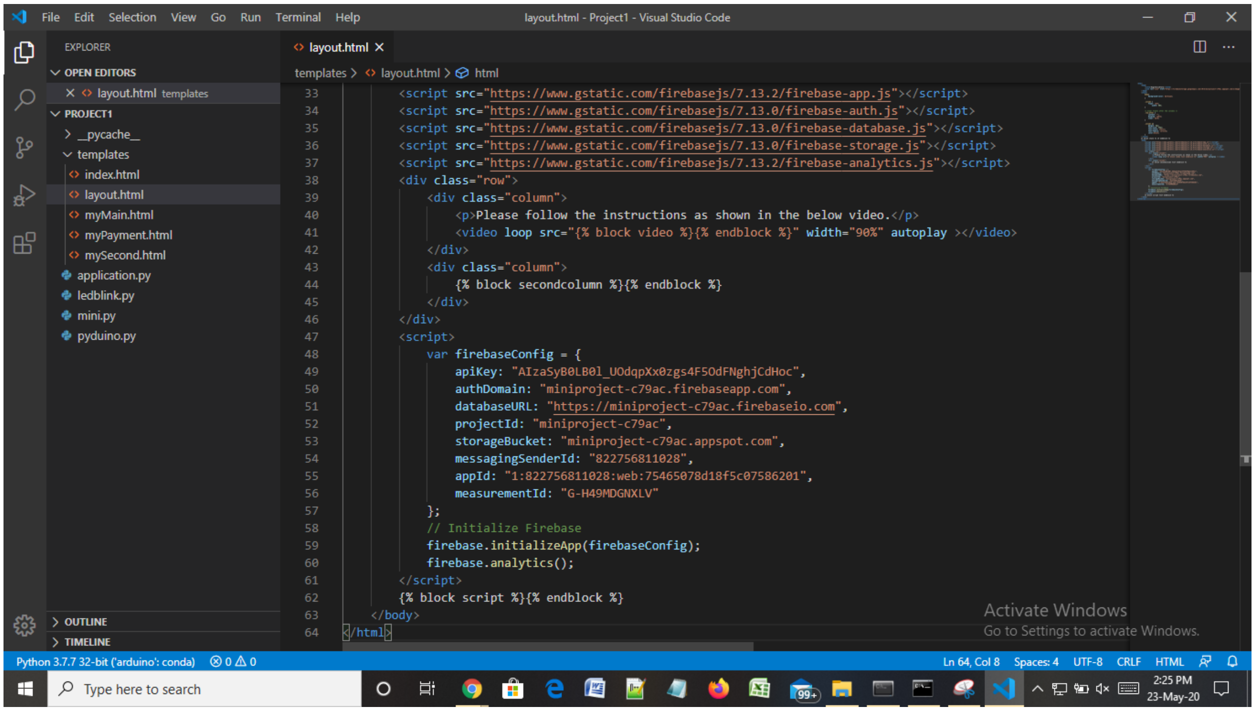
```
    <body>
        <div class="main_img">
            <img src="https://firebasestorage.googleapis.com/v0/b/miniproject-c79ac.appspot.com/o/Images%2FSIT-T
        </div>
        <div class="main_class">
            <h1>WELCOME TO</h1>
            <h3>DIGITAL LIBRARY!</h3>
            <p>To proceed further please
                <a href="mySecond.html"><button type="button">Click</button></a>
            </p>
        </div>
    </body>
</html>
```

(d) layout.html module –

Generally, we always prefer to reduce repetitive works. In this project book insertion module and money insertion module have the same template so in order to not to code again jinja2 provides us with a fascinating feature of extending layout any number of times i.e. write once use many times.

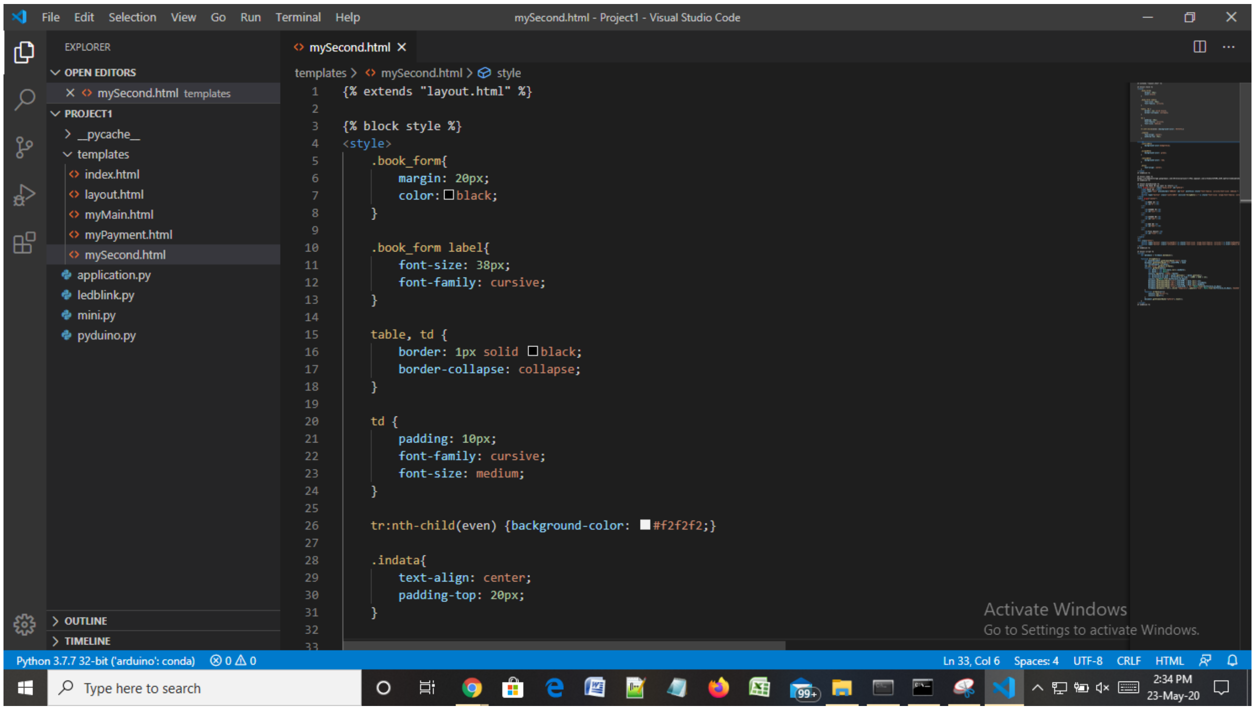

File Edit Selection View Go Run Terminal Help
layout.html - Project1 - Visual Studio Code
EXPLORER
OPEN EDITORS
layout.html templates
PROJECT1
pycache
templates
index.html
layout.html
myMain.html
myPayment.html
mySecond.html
application.py
ledblink.py
mini.py
pyduino.py
OUTLINE
TIMELINE
layout.html
templates > layout.html > html > style > .column
Activate Windows
Go to Settings to activate Windows.
Python 3.7.7 32-bit ('arduino': conda) ⊗ 0 ⚠ 0
Ln 11, Col 18 Spaces: 4 UTF-8 CRLF HTML
Type here to search
2:24 PM
23-May-20

```
<html>
    <head>
        <title>DigitalLibrary</title>
        <link rel="icon" href="https://firebasestorage.googleapis.com/v0/b/miniproject-c79ac.appspot.com/o/Image
    </head>
    <style>
        body{
            background-color: darkcyan;
        }

        .column {
            float: left;
                width: 50%;
        }

        /* Clear floats after the columns */
        .row:after {
            content: "";
            display: table;
            clear: both;
        }

        .column p{
            margin: 5px;
            font-size: 35px;
            font-style: italic;
            font-family: cursive;
            color: black;
        }
    </style>
    {% block style %} {% endblock %}
    <body>
        <script src="https://www.gstatic.com/firebasejs/7.13.2/firebase-app.js"></script>
```

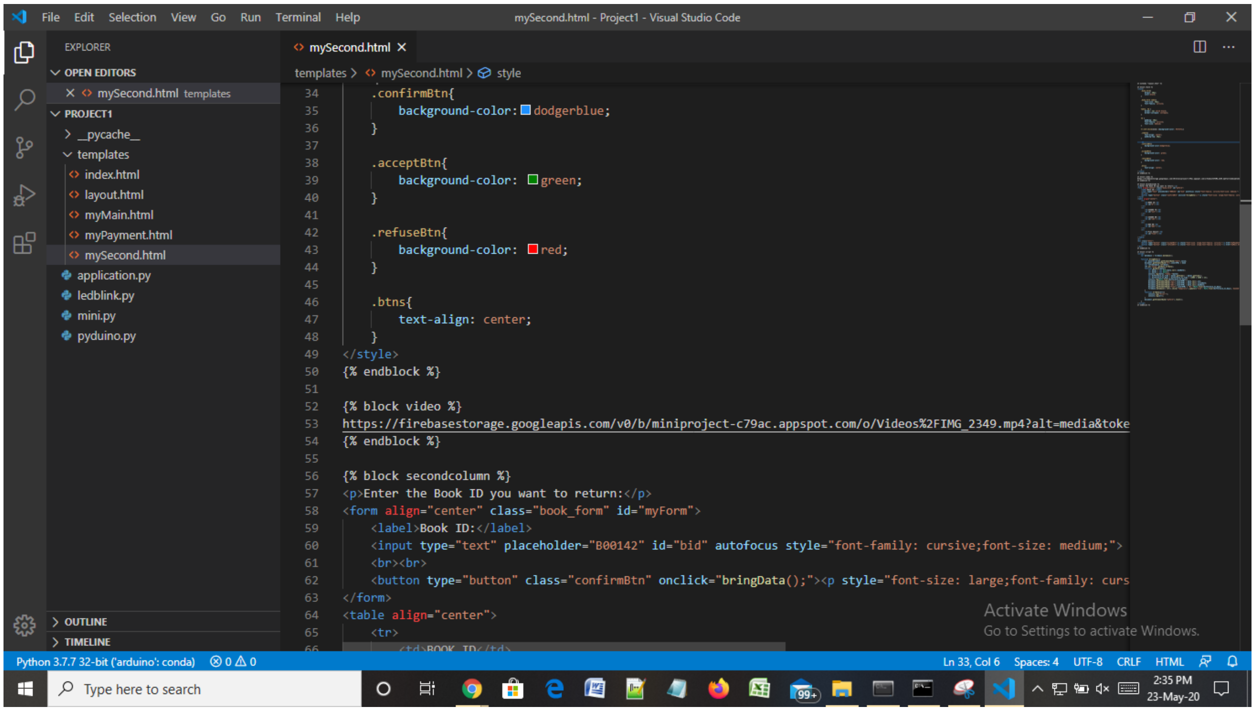


```
        <script src="https://www.gstatic.com/firebasejs/7.13.2/firebase-app.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.13.0/firebase-auth.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.13.0/firebase-database.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.13.0/firebase-storage.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.13.2/firebase-analytics.js"></script>
        <div class="row">
            <div class="column">
                <p>Please follow the instructions as shown in the below video.</p>
                <video loop src="{% block video %}{% endblock %}" width="90%" autoplay ></video>
            </div>
            <div class="column">
                {% block secondcolumn %}{% endblock %}
            </div>
        </div>
        <script>
            var firebaseConfig = {
                apiKey: "AIzaSyB0LB0l_UOdqpXx0zgs4F5OdFNghjCdHoc",
                authDomain: "miniproject-c79ac.firebaseapp.com",
                databaseURL: "https://miniproject-c79ac.firebaseio.com",
                projectId: "miniproject-c79ac",
                storageBucket: "miniproject-c79ac.appspot.com",
                messagingSenderId: "822756811028",
                appId: "1:822756811028:web:75465078d18f5c07586201",
                measurementId: "G-H49MDGNXLV"
            };
            // Initialize Firebase
            firebase.initializeApp(firebaseConfig);
            firebase.analytics();
        </script>
        {% block script %}{% endblock %}
    </body>
</html>
```

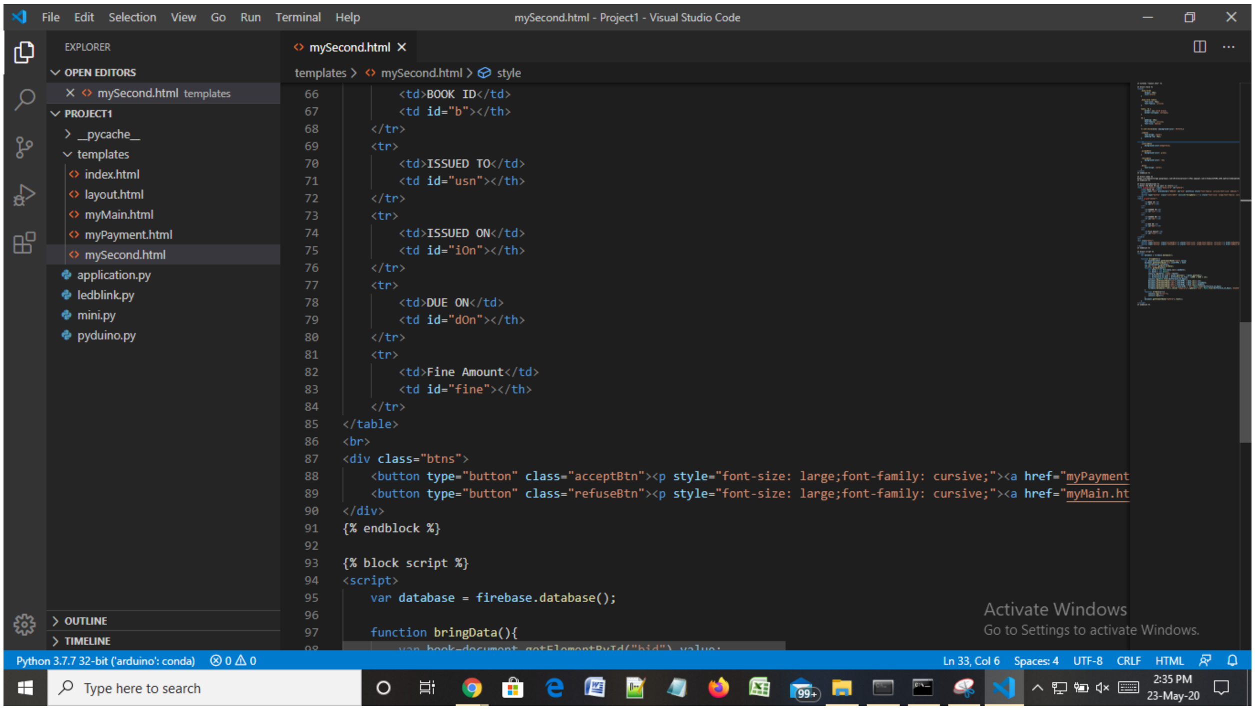
(e) Book id, details retrieval module –



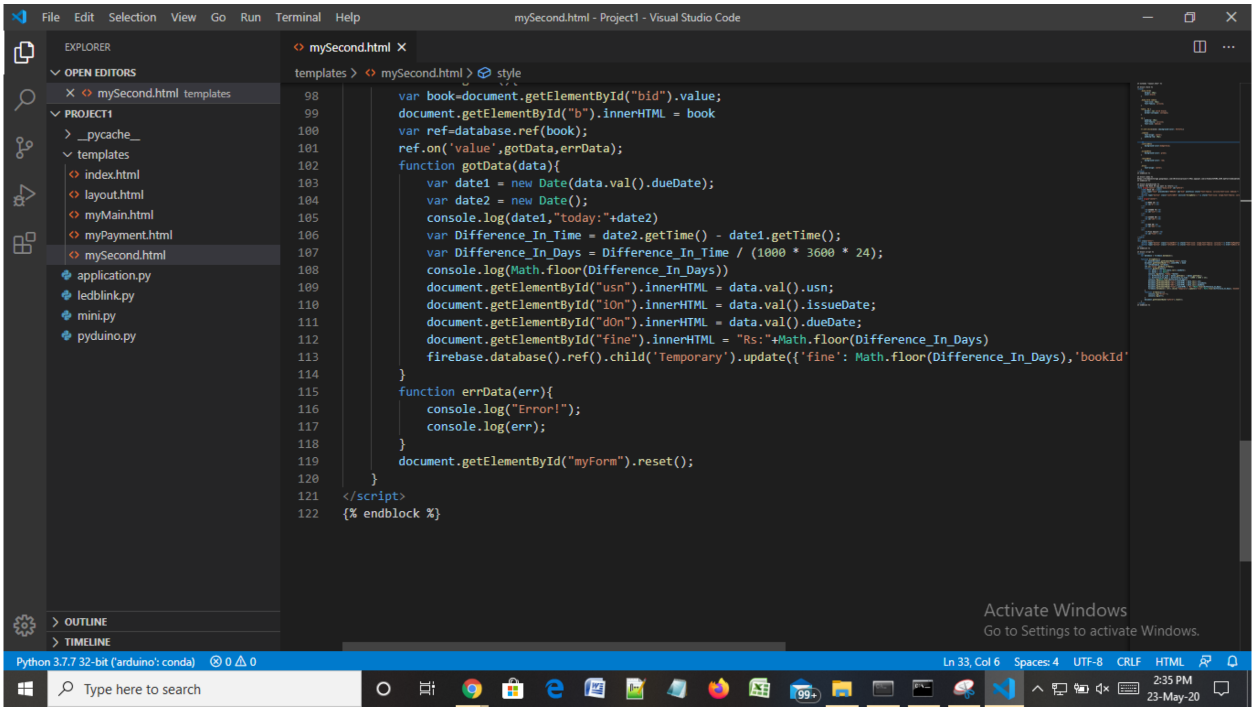
```
{% extends "layout.html" %}

{% block style %}
<style>
    .book_form{
        margin: 20px;
        color: □black;
    }

    .book_form label{
        font-size: 38px;
        font-family: cursive;
    }

    table, td {
        border: 1px solid □black;
        border-collapse: collapse;
    }

    td {
        padding: 10px;
        font-family: cursive;
        font-size: medium;
    }

    tr:nth-child(even) {background-color: □#f2f2f2;}

    .indata{
        text-align: center;
        padding-top: 20px;
    }

```



```
    .confirmBtn{
        background-color: dodgerblue;
    }

    .acceptBtn{
        background-color: green;
    }

    .refuseBtn{
        background-color: red;
    }

    .btns{
        text-align: center;
    }
</style>
{% endblock %}

{% block video %}
https://firebasestorage.googleapis.com/v0/b/miniproject-c79ac.appspot.com/o/Videos%2FIMG_2349.mp4?alt=media&toke
{% endblock %}

{% block secondcolumn %}
<p>Enter the Book ID you want to return:</p>
<form align="center" class="book_form" id="myForm">
    <label>Book ID:</label>
    <input type="text" placeholder="B00142" id="bid" autofocus style="font-family: cursive;font-size: medium;">
    <br><br>
    <button type="button" class="confirmBtn" onclick="bringData();"><p style="font-size: large;font-family: curs
</form>
<table align="center">
    <tr>
        <td>BOOK ID</td>
```

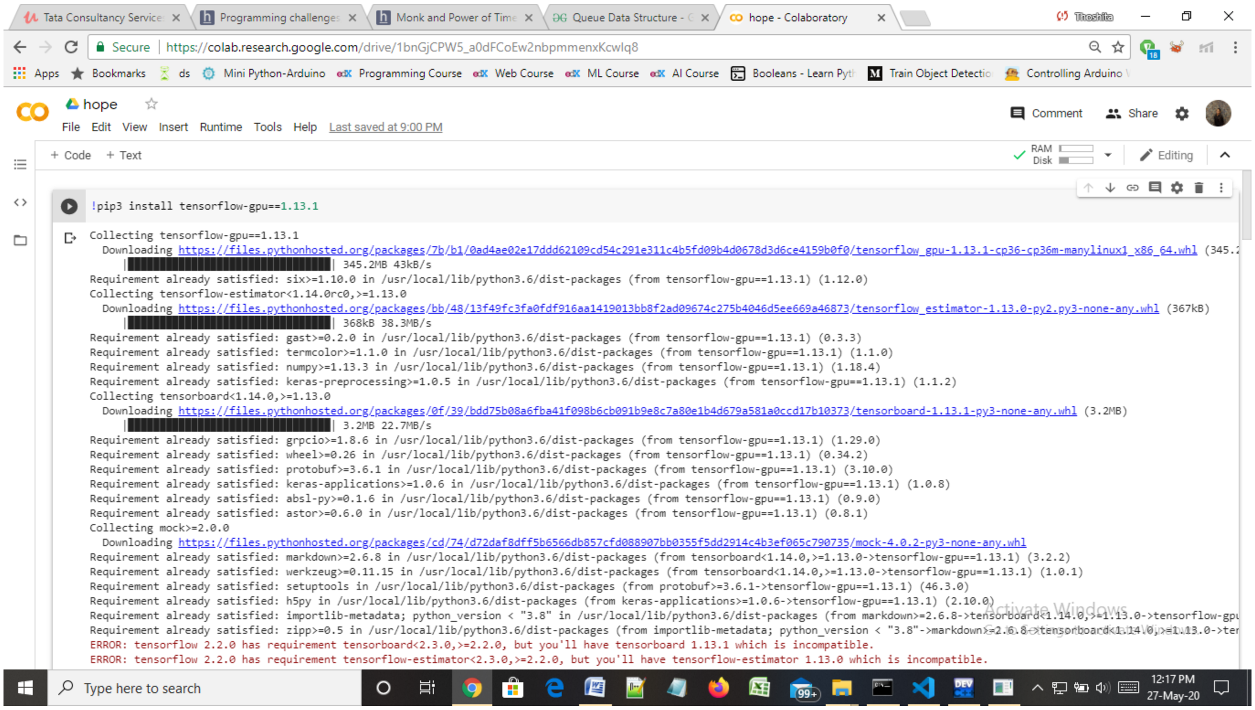


```
        <td>BOOK ID</td>
        <td id="b"></th>
    </tr>
    <tr>
        <td>ISSUED TO</td>
        <td id="usn"></th>
    </tr>
    <tr>
        <td>ISSUED ON</td>
        <td id="iOn"></th>
    </tr>
    <tr>
        <td>DUE ON</td>
        <td id="dOn"></th>
    </tr>
    <tr>
        <td>Fine Amount</td>
        <td id="fine"></th>
    </tr>
</table>
<br>
<div class="btns">
    <button type="button" class="acceptBtn"><p style="font-size: large;font-family: cursive;"><a href="myPayment
    <button type="button" class="refuseBtn"><p style="font-size: large;font-family: cursive;"><a href="myMain.ht
</div>
{% endblock %}

{% block script %}
<script>
    var database = firebase.database();

    function bringData(){
        var book=document.getElementById("bid").value;
```



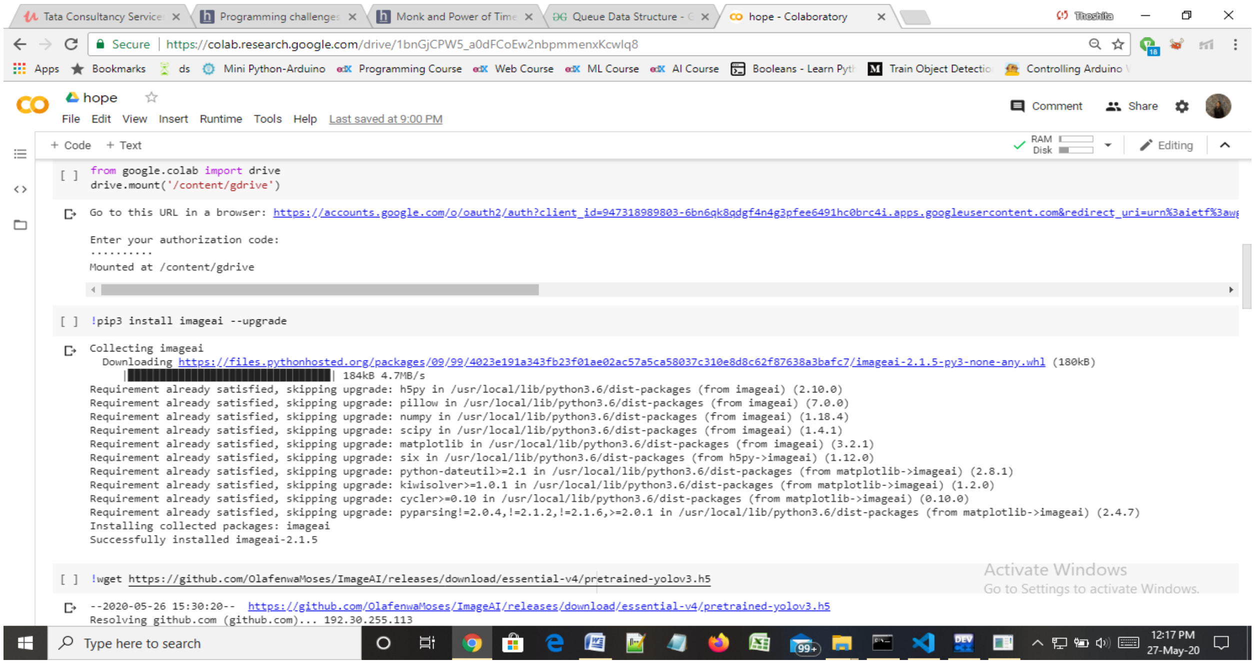
```
            var book=document.getElementById("bid").value;
            document.getElementById("b").innerHTML = book
            var ref=database.ref(book);
            ref.on('value',gotData,errData);
            function gotData(data){
                var date1 = new Date(data.val().dueDate);
                var date2 = new Date();
                console.log(date1,"today:"+date2)
                var Difference_In_Time = date2.getTime() - date1.getTime();
                var Difference_In_Days = Difference_In_Time / (1000 * 3600 * 24);
                console.log(Math.floor(Difference_In_Days))
                document.getElementById("usn").innerHTML = data.val().usn;
                document.getElementById("iOn").innerHTML = data.val().issueDate;
                document.getElementById("dOn").innerHTML = data.val().dueDate;
                document.getElementById("fine").innerHTML = "Rs:"+Math.floor(Difference_In_Days)
                firebase.database().ref().child('Temporary').update({'fine': Math.floor(Difference_In_Days),'bookId'
            }
            function errData(err){
                console.log("Error!");
                console.log(err);
            }
            document.getElementById("myForm").reset();
        }
    </script>
    {% endblock %}
```

IMPLEMENTATION OF IMAGE PROCESSING using Google Collaboratory:


Tata Consultancy Service
Programming challenges
Monk and Power of Time
Queue Data Structure
hope - Colaboratory
Secure https://colab.research.google.com/drive/1bnGjCPW5_a0dFCoEw2nbpmmenxKcwlq8
Apps Bookmarks ds Mini Python-Arduino Programming Course Web Course ML Course AI Course Booleans - Learn Pyt Train Object Detectio Controlling Arduino
hope
File Edit View Insert Runtime Tools Help Last saved at 9:00 PM
Comment Share
+ Code + Text
RAM Disk
Editing
Type here to search
12:17 PM
27-May-20

```
!pip3 install tensorflow-gpu==1.13.1
```

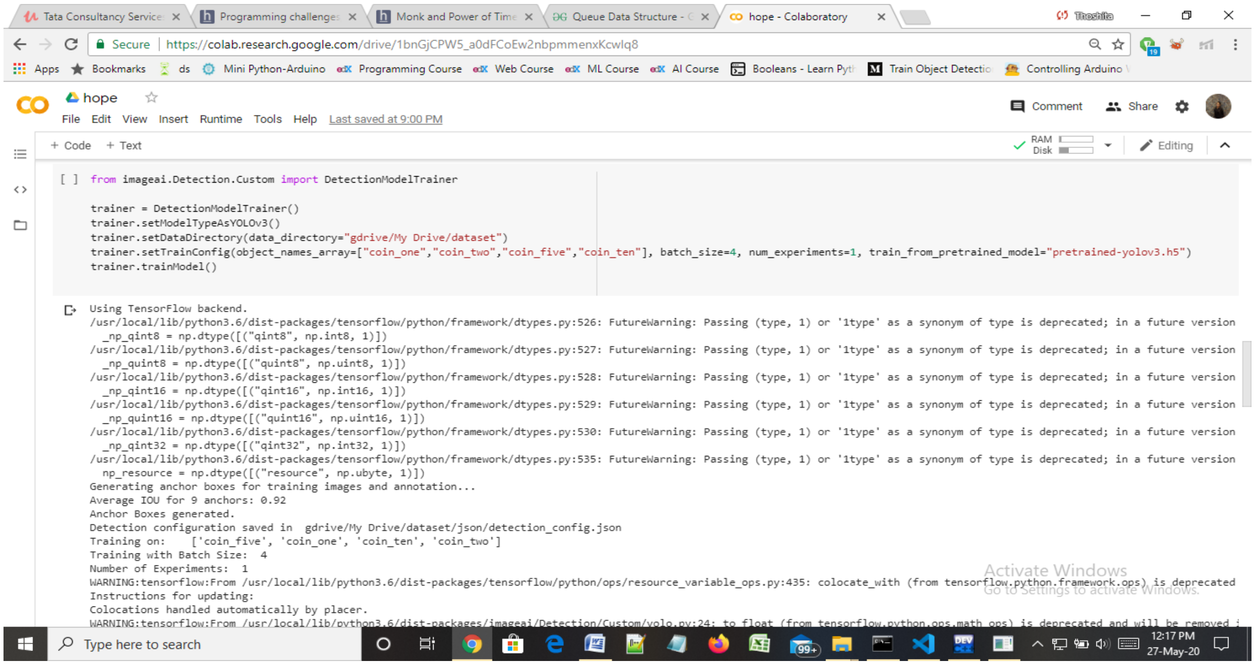
```
Collecting tensorflow-gpu==1.13.1
  Downloading https://files.pythonhosted.org/packages/7b/b1/0ad4ae02e17ddd62109cd54c291e311c4b5fd09b4d0678d3d6ce4159b0f0/tensorflow_gpu-1.13.1-cp36-cp36m-manylinux1_x86_64.whl (345.2
     |████████████████████████████████| 345.2MB 43kB/s
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (1.12.0)
Collecting tensorflow-estimator<1.14.0rc0,>=1.13.0
  Downloading https://files.pythonhosted.org/packages/bb/48/13f49fc3fa0fdf916aa1419013bb8f2ad09674c275b4046d5ee669a46873/tensorflow_estimator-1.13.0-py2.py3-none-any.whl (367kB)
     |████████████████████████████████| 368kB 38.3MB/s
Requirement already satisfied: gast>=0.2.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (0.3.3)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (1.1.0)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (1.18.4)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (1.1.2)
Collecting tensorboard<1.14.0,>=1.13.0
  Downloading https://files.pythonhosted.org/packages/0f/39/bdd75b08a6fba41f098b6cb091b9e8c7a80e1b4d679a581a0ccd17b10373/tensorboard-1.13.1-py3-none-any.whl (3.2MB)
     |████████████████████████████████| 3.2MB 22.7MB/s
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (1.29.0)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (0.34.2)
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (3.10.0)
Requirement already satisfied: keras-applications>=1.0.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (1.0.8)
Requirement already satisfied: absl-py>=0.1.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (0.9.0)
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.13.1) (0.8.1)
Collecting mock>=2.0.0
  Downloading https://files.pythonhosted.org/packages/cd/74/d72daf8dff5b6566db857cfd088907bb0355f5dd2914c4b3ef065c790735/mock-4.0.2-py3-none-any.whl
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-packages (from tensorboard<1.14.0,>=1.13.0->tensorflow-gpu==1.13.1) (3.2.2)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.6/dist-packages (from tensorboard<1.14.0,>=1.13.0->tensorflow-gpu==1.13.1) (1.0.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (from protobuf>=3.6.1->tensorflow-gpu==1.13.1) (46.3.0)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from keras-applications>=1.0.6->tensorflow-gpu==1.13.1) (2.10.0)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in /usr/local/lib/python3.6/dist-packages (from markdown>=2.6.8->tensorboard<1.14.0,>=1.13.0->tensorflow-gpu
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-packages (from importlib-metadata; python_version < "3.8"->markdown>=2.6.8->tensorboard<1.14.0,>=1.13.0->ter
ERROR: tensorflow 2.2.0 has requirement tensorboard<2.3.0,>=2.2.0, but you'll have tensorboard 1.13.1 which is incompatible.
ERROR: tensorflow 2.2.0 has requirement tensorflow-estimator<2.3.0,>=2.2.0, but you'll have tensorflow-estimator 1.13.0 which is incompatible.
```


Tata Consultancy Service
Programming challenge
Monk and Power of Time
Queue Data Structure -
hope - Colaboratory
Secure | https://colab.research.google.com/drive/1bnGjCPW5_a0dFCoEw2nbpmmenxKcwlq8
Apps Bookmarks ds Mini Python-Arduino Programming Course Web Course ML Course AI Course Booleans - Learn Pyth Train Object Detectio Controlling Arduino
hope
File Edit View Insert Runtime Tools Help Last saved at 9:00 PM
Comment Share
+ Code + Text
RAM Disk
Editing
Activate Windows
Go to Settings to activate Windows.
Type here to search
12:17 PM
27-May-20

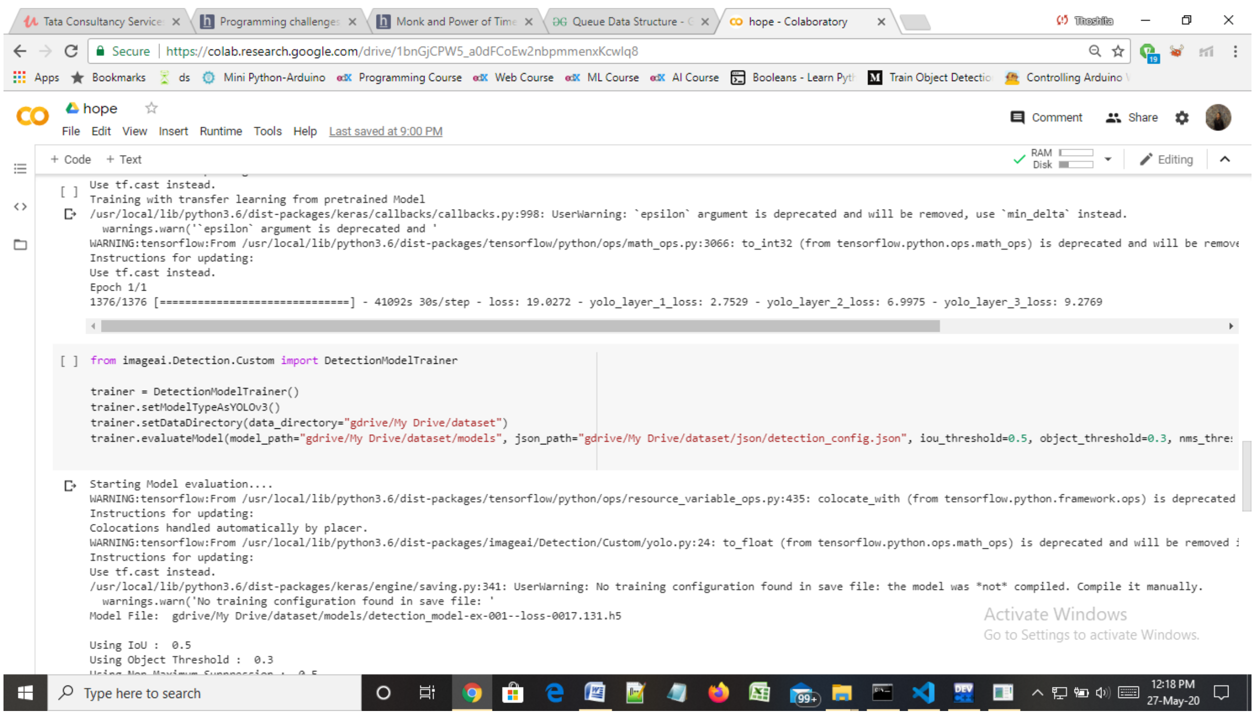
```
[ ] from google.colab import drive
    drive.mount('/content/gdrive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg

Enter your authorization code:
··········
Mounted at /content/gdrive
```

```
[ ] !pip3 install imageai --upgrade

Collecting imageai
  Downloading https://files.pythonhosted.org/packages/09/99/4023e191a343fb23f01ae02ac57a5ca58037c310e8d8c62f87638a3bafc7/imageai-2.1.5-py3-none-any.whl (180kB)
     |████████████████████████████████| 184kB 4.7MB/s
Requirement already satisfied, skipping upgrade: h5py in /usr/local/lib/python3.6/dist-packages (from imageai) (2.10.0)
Requirement already satisfied, skipping upgrade: pillow in /usr/local/lib/python3.6/dist-packages (from imageai) (7.0.0)
Requirement already satisfied, skipping upgrade: numpy in /usr/local/lib/python3.6/dist-packages (from imageai) (1.18.4)
Requirement already satisfied, skipping upgrade: scipy in /usr/local/lib/python3.6/dist-packages (from imageai) (1.4.1)
Requirement already satisfied, skipping upgrade: matplotlib in /usr/local/lib/python3.6/dist-packages (from imageai) (3.2.1)
Requirement already satisfied, skipping upgrade: six in /usr/local/lib/python3.6/dist-packages (from h5py->imageai) (1.12.0)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->imageai) (2.8.1)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->imageai) (1.2.0)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /usr/local/lib/python3.6/dist-packages (from matplotlib->imageai) (0.10.0)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->imageai) (2.4.7)
Installing collected packages: imageai
Successfully installed imageai-2.1.5
```

```
[ ] !wget https://github.com/OlafenwaMoses/ImageAI/releases/download/essential-v4/pretrained-yolov3.h5

--2020-05-26 15:30:20--  https://github.com/OlafenwaMoses/ImageAI/releases/download/essential-v4/pretrained-yolov3.h5
Resolving github.com (github.com)... 192.30.255.113
```


Tata Consultancy Service
Programming challenge
Monk and Power of Time
Queue Data Structure -
hope - Colaboratory
Secure | https://colab.research.google.com/drive/1bnGjCPW5_a0dFCoEw2nbpmmenxKcwlq8
Apps Bookmarks ds Mini Python-Arduino Programming Course Web Course ML Course AI Course Booleans - Learn Pyth Train Object Detectio Controlling Arduino
hope
File Edit View Insert Runtime Tools Help Last saved at 9:00 PM
Comment Share
+ Code + Text
RAM Disk
Editing
Activate Windows
Go to Settings to activate Windows.
Type here to search
12:17 PM
27-May-20

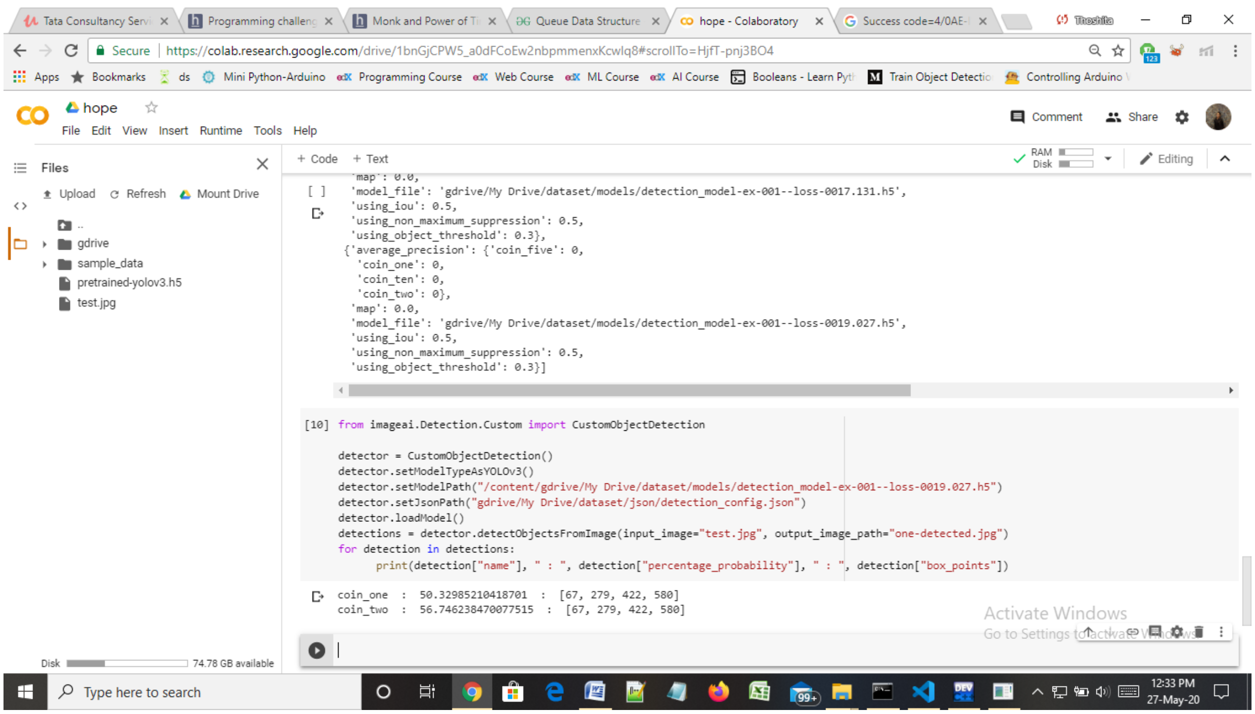
```
[ ] from imageai.Detection.Custom import DetectionModelTrainer

    trainer = DetectionModelTrainer()
    trainer.setModelTypeAsYOLOv3()
    trainer.setDataDirectory(data_directory="gdrive/My Drive/dataset")
    trainer.setTrainConfig(object_names_array=["coin_one","coin_two","coin_five","coin_ten"], batch_size=4, num_experiments=1, train_from_pretrained_model="pretrained-yolov3.h5")
    trainer.trainModel()

Using TensorFlow backend.
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version
  np_resource = np.dtype([("resource", np.ubyte, 1)])
Generating anchor boxes for training images and annotation...
Average IOU for 9 anchors: 0.92
Anchor Boxes generated.
Detection configuration saved in  gdrive/My Drive/dataset/json/detection_config.json
Training on:    ['coin_five', 'coin_one', 'coin_ten', 'coin_two']
Training with Batch Size:  4
Number of Experiments:  1
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_ops.py:435: colocate_with (from tensorflow.python.framework.ops) is deprecated
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/imageai/Detection/Custom/yolo.py:24: to_float (from tensorflow.python.ops.math_ops) is deprecated and will be removed i
```



```
Use tf.cast instead.
Training with transfer learning from pretrained Model
/usr/local/lib/python3.6/dist-packages/keras/callbacks/callbacks.py:998: UserWarning: `epsilon` argument is deprecated and will be removed, use `min_delta` instead.
  warnings.warn('`epsilon` argument is deprecated and '
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be remove
Instructions for updating:
Use tf.cast instead.
Epoch 1/1
1376/1376 [==============================] - 41092s 30s/step - loss: 19.0272 - yolo_layer_1_loss: 2.7529 - yolo_layer_2_loss: 6.9975 - yolo_layer_3_loss: 9.2769
```

```
from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="gdrive/My Drive/dataset")
trainer.evaluateModel(model_path="gdrive/My Drive/dataset/models", json_path="gdrive/My Drive/dataset/json/detection_config.json", iou_threshold=0.5, object_threshold=0.3, nms_thre:
```

```
Starting Model evaluation....
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_ops.py:435: colocate_with (from tensorflow.python.framework.ops) is deprecated
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/imageai/Detection/Custom/yolo.py:24: to_float (from tensorflow.python.ops.math_ops) is deprecated and will be removed i
Instructions for updating:
Use tf.cast instead.
/usr/local/lib/python3.6/dist-packages/keras/engine/saving.py:341: UserWarning: No training configuration found in save file: the model was *not* compiled. Compile it manually.
  warnings.warn('No training configuration found in save file: '
Model File:  gdrive/My Drive/dataset/models/detection_model-ex-001--loss-0017.131.h5

Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
```



```
 'map': 0.0,
 'model_file': 'gdrive/My Drive/dataset/models/detection_model-ex-001--loss-0017.131.h5',
 'using_iou': 0.5,
 'using_non_maximum_suppression': 0.5,
 'using_object_threshold': 0.3},
{'average_precision': {'coin_five': 0,
  'coin_one': 0,
  'coin_ten': 0,
  'coin_two': 0},
 'map': 0.0,
 'model_file': 'gdrive/My Drive/dataset/models/detection_model-ex-001--loss-0019.027.h5',
 'using_iou': 0.5,
 'using_non_maximum_suppression': 0.5,
 'using_object_threshold': 0.3}]
```

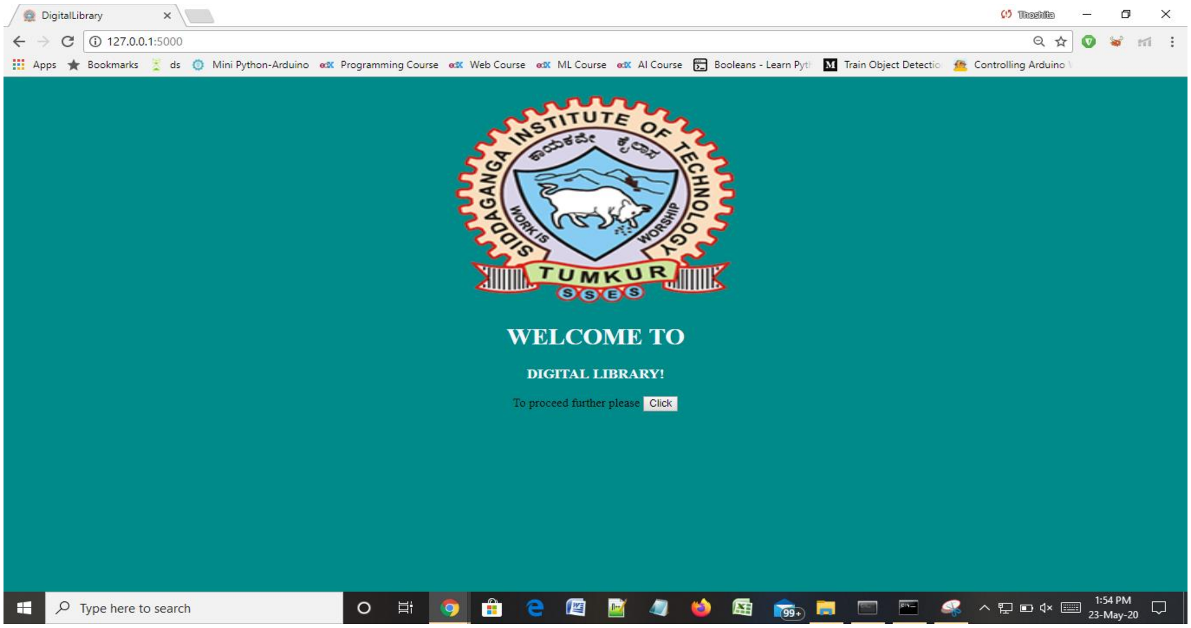
```
[10] from imageai.Detection.Custom import CustomObjectDetection

     detector = CustomObjectDetection()
     detector.setModelTypeAsYOLOv3()
     detector.setModelPath("/content/gdrive/My Drive/dataset/models/detection_model-ex-001--loss-0019.027.h5")
     detector.setJsonPath("gdrive/My Drive/dataset/json/detection_config.json")
     detector.loadModel()
     detections = detector.detectObjectsFromImage(input_image="test.jpg", output_image_path="one-detected.jpg")
     for detection in detections:
           print(detection["name"], " : ", detection["percentage_probability"], " : ", detection["box_points"])
```

```
coin_one  :  50.32985210418701  :  [67, 279, 422, 580]
coin_two  :  56.746238470077515  :  [67, 279, 422, 580]
```

CHAPTER 5

RESULTS

The application has been built & the working flow for the user is described below through screen-shots as shown in figure 5.1.

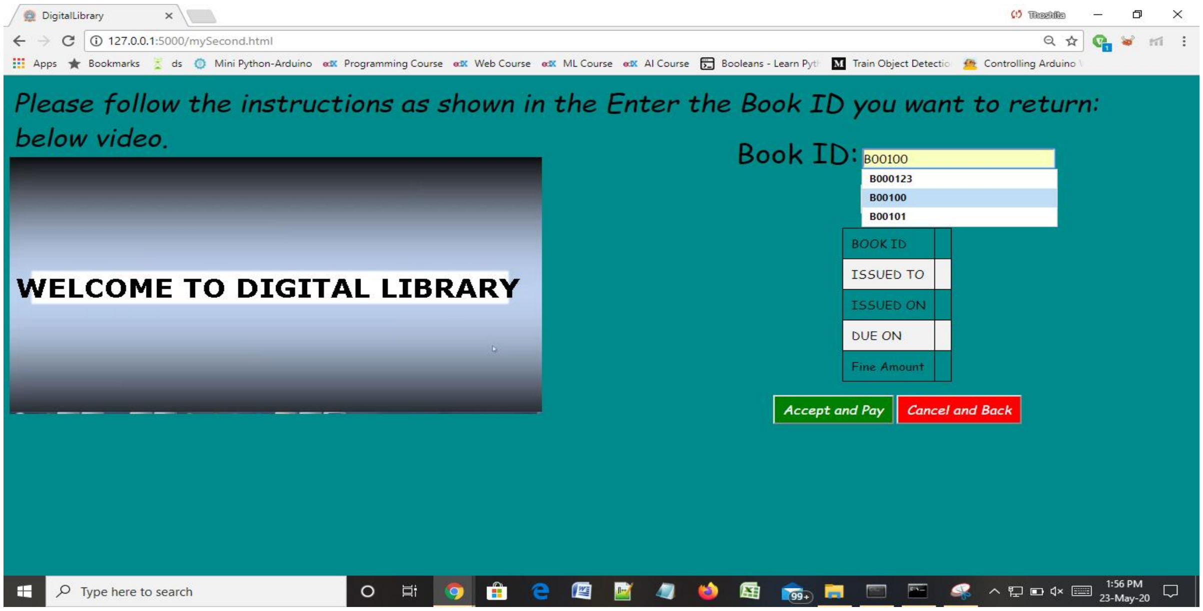


WELCOME TO

DIGITAL LIBRARY!

To proceed further please Click

5.1(a) Main Screen



Please follow the instructions as shown in the below video.

WELCOME TO DIGITAL LIBRARY

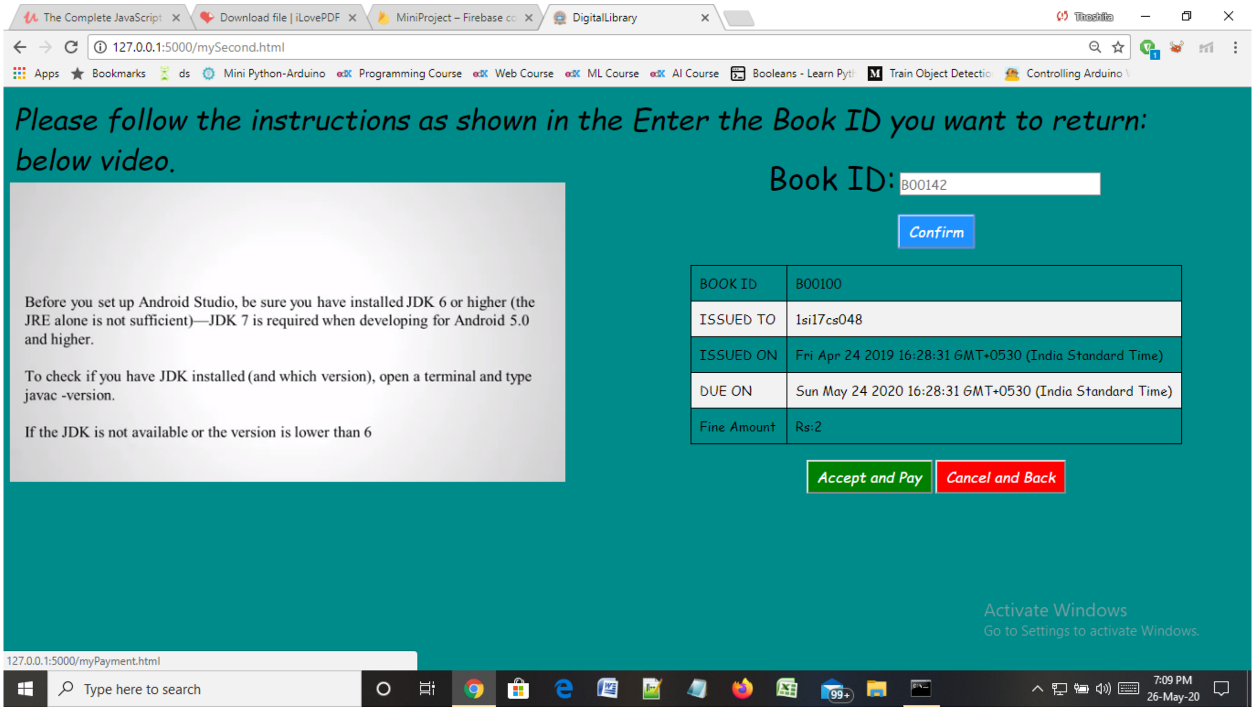
Enter the Book ID you want to return:

Book ID: B00100

B000123
B00100
B00101

BOOK ID	
ISSUED TO	
ISSUED ON	
DUE ON	
Fine Amount	

5.1(b) Second Screen (input of book id & tutorial video playing at the back for supporting users in how to insert)



Please follow the instructions as shown in the below video.

Before you set up Android Studio, be sure you have installed JDK 6 or higher (the JRE alone is not sufficient)—JDK 7 is required when developing for Android 5.0 and higher.

To check if you have JDK installed (and which version), open a terminal and type javac -version.

If the JDK is not available or the version is lower than 6

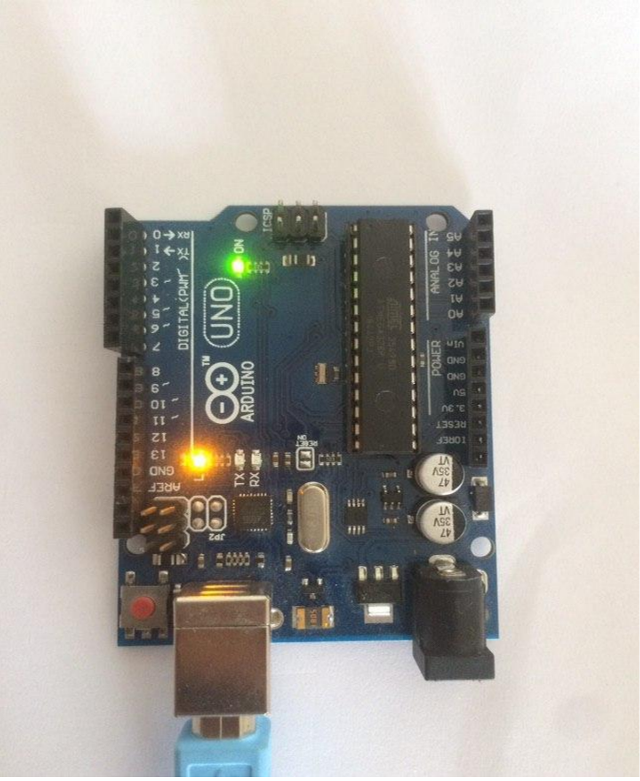
Enter the Book ID you want to return:

Book ID: B00142

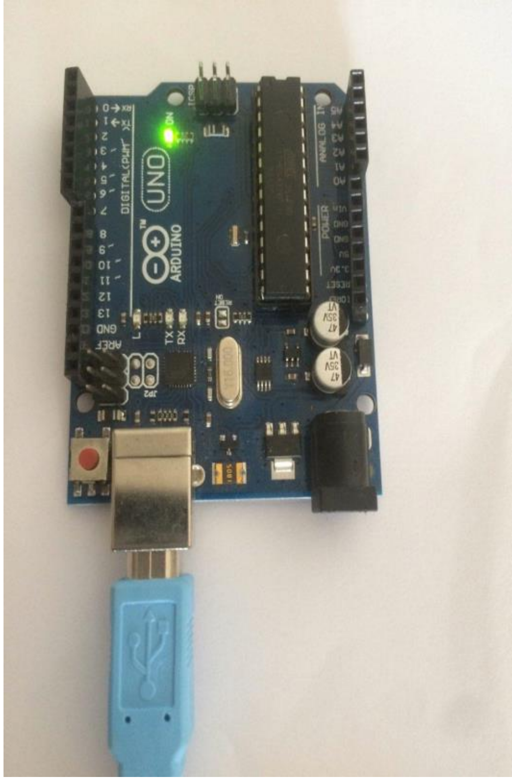
BOOK ID	B00100
ISSUED TO	1si17cs048
ISSUED ON	Fri Apr 24 2019 16:28:31 GMT+0530 (India Standard Time)
DUE ON	Sun May 24 2020 16:28:31 GMT+0530 (India Standard Time)
Fine Amount	Rs:2

5.1(c) second screen (after book id entry verifying book details in db and retrieving them)

RESULTS OF ARDUINO FUNCTIONING

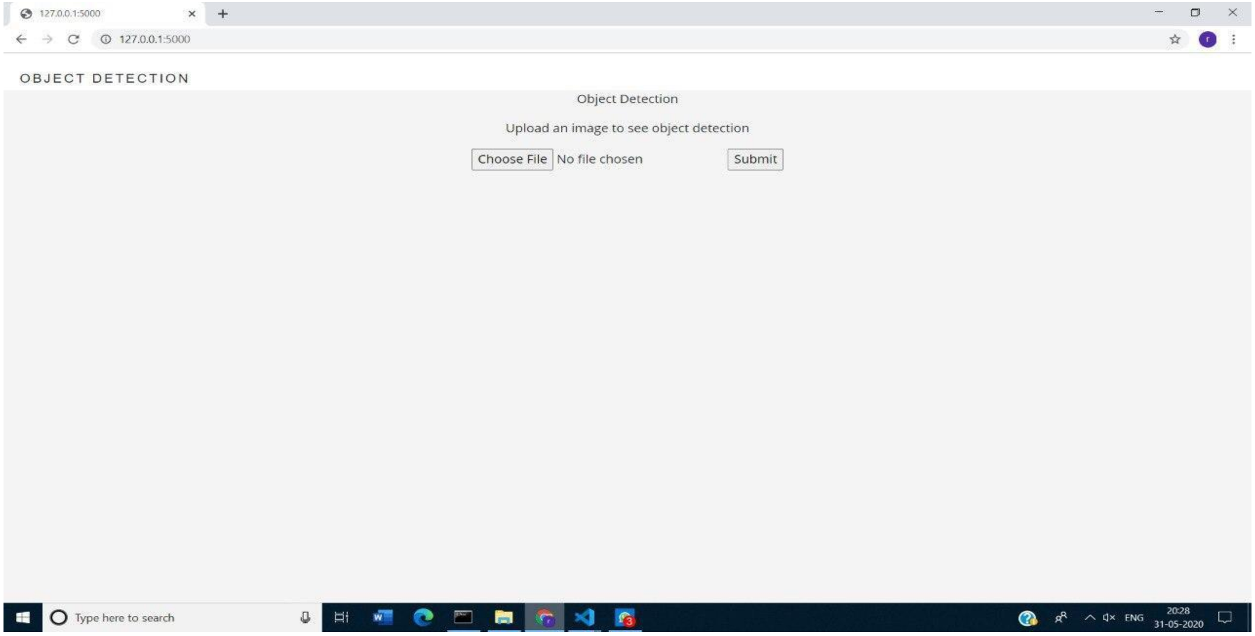


5.2(a) Built-in LED in ON state



5.2(b) Built-in LED in OFF state

RESULTS OF IMAGE PROCESSING

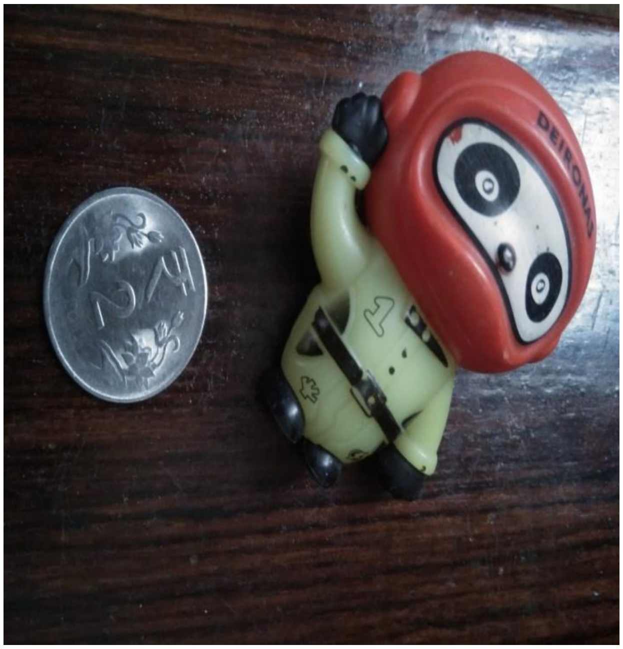


OBJECT DETECTION

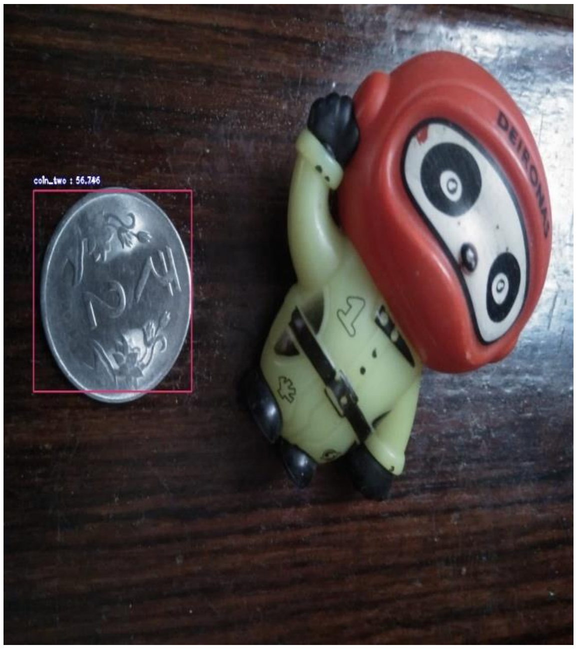
Object Detection

Upload an image to see object detection

No file chosen



5.3(a) Input image at money insertion port



5.3(b) Output – detection of Rs.2

CHAPTER 6

CONCLUSION

It was a wonderful experience for us while working on this project. This project took us through the various phases of project development and gave us the real insight into the world of software engineering. The goal of the project was to minimize the manual efforts being put during returning the books as well as to reduce the waiting time in long queues. In future implementation we are planning to improve the fine collection by interfacing a payment gateway so that even when the machine lacks with proper change to return that wouldn’t lead to another problem.

CHAPTER 7

REFERENCES

[1] www.wikipedia.org

[2] Train Object Detection AI -

https://medium.com/deepquestai/train-object-detection-ai-with-6-lines-of-code-6d087063f6ff

[3] Web Development using Python – https://courses.edx.org/

[4] Controlling Arduino with Python based Web API –

https://www.instructables.com/id/Controlling-Arduino-with-python-based-web-API-No-p/

[5] Object Detection using Tensorflow –

https://www.edureka.co/blog/tensorflow-object-detection-tutorial/

[6] Creating your own object detector –

https://towardsdatascience.com/creating-your-own-object-detector-ad69dda69c85

[7] Indian currency, Indian coins – www.kaggle.com