Implementation

4.1 Introduction

This chapter will contain the implementation of our project, to implement our project after our analysis, first thing to start with implementing the Login Form (figure1).

4.2 Login Form:

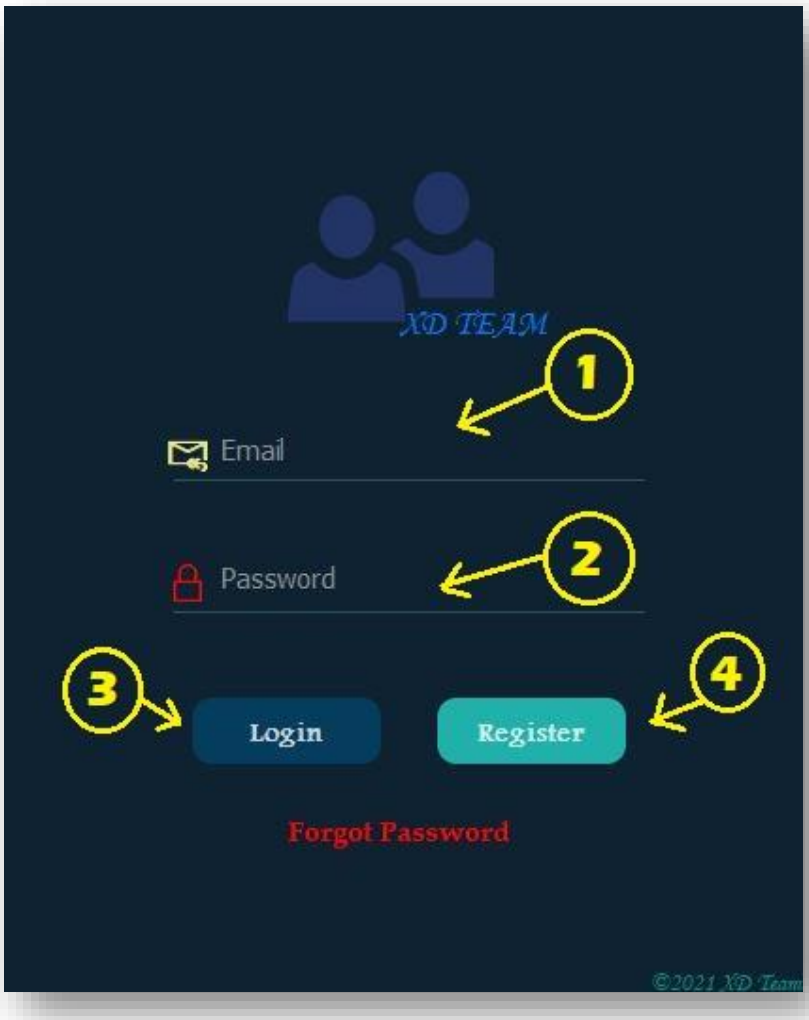

Figure 1- Login Form.

4.3 Consisting of:-

1. Email -To take email from user.
2. Password -To take password from user.
3. Login button -To check the name and password.
4. Register button -To add a new user.
5. Forgot password -To reset user.

4.4 Login button:-

If user pressed the Login button, this code will be executed (figure 2).

```
def login(self):
    EMAIL = self.email.text()
    PASSWORD = self.password.text()
    #to connect with database .
    con =mysql.connector.connect(user='root',password='ammar45',host='localhost',database='user')
    #Create a Cursor object to execute queries.
    manager = con.cursor()
    get_email = ("SELECT email FROM users WHERE email = '%s'"%str(EMAIL))
    get_password = ("SELECT pass FROM users WHERE pass = '%s'"%str(PASSWORD))
    manager.execute(get_email,get_password)
    if self.email.text()== get_email and self.password.text() == get_password:
        self.refreshAll()
        ui.hide()  # hide the main window
        self.outputWindow_()  # Create and open new output window
    else:
        msg = QMessageBox()
        msg.setText("check your email or password")
        msg.exec_()
```

Figure 2 - Execute The Code Login Form.

4.5 Register button:-

If user pressed the registration button, the registration form will open to add user data (figure 3).

4.6 Registration form:

Figure 3 - Registration Form.

4.7 Consisting of:

1. First name-This field will take first name from user.
2. Last name-This field will take last name from user.
3. Email- This field will take email from user.
4. Password- This field will take password from user.
5. Confirm password- This field will take confirm password from user.
6. Register button – This button will take all fields and added them in the database.

4.8 Register button:

If user pressed the Register button, this code will be executed (figure 4).

```python
def signup(self):
    try:
        con =mysql.connector.connect(user='root',password='ammar45',host='localhost',database='user')
        manager   = con.cursor()
        first_name= self.fname.text()
        last_name =  self.lname.text()
        email     =  self.email.text()
        password1 =  self.pass1.text()
        password2 =  self.pass2.text()
        if (password1 != password2):
            msgBox = QMessageBox()
            msgBox.setText("cheek yor password")
            msgBox.exec_()
        else:
            query = "INSERT INTO users (fname,lname,email, pass) VALUES (%s, %s, %s, %s)"
            value = (first_name,last_name,email, password1)
            manager.execute(query, value)
            con.commit()

            self.fname.setText('')
            self.lname.setText('')
            self.email.setText('')
            self.pass1.setText('')
            self.pass2.setText('')
            msgBox = QMessageBox()
            msgBox.setText("Data Inserted")
            msgBox.exec_()
    except:
        msgBox = QMessageBox()
        msgBox.setText("error")
        msgBox.exec_()
```

Figure 4 - Execute The Code Registration Form.

4.9 Student attendance:

After Login, this form will appear (figure 5).

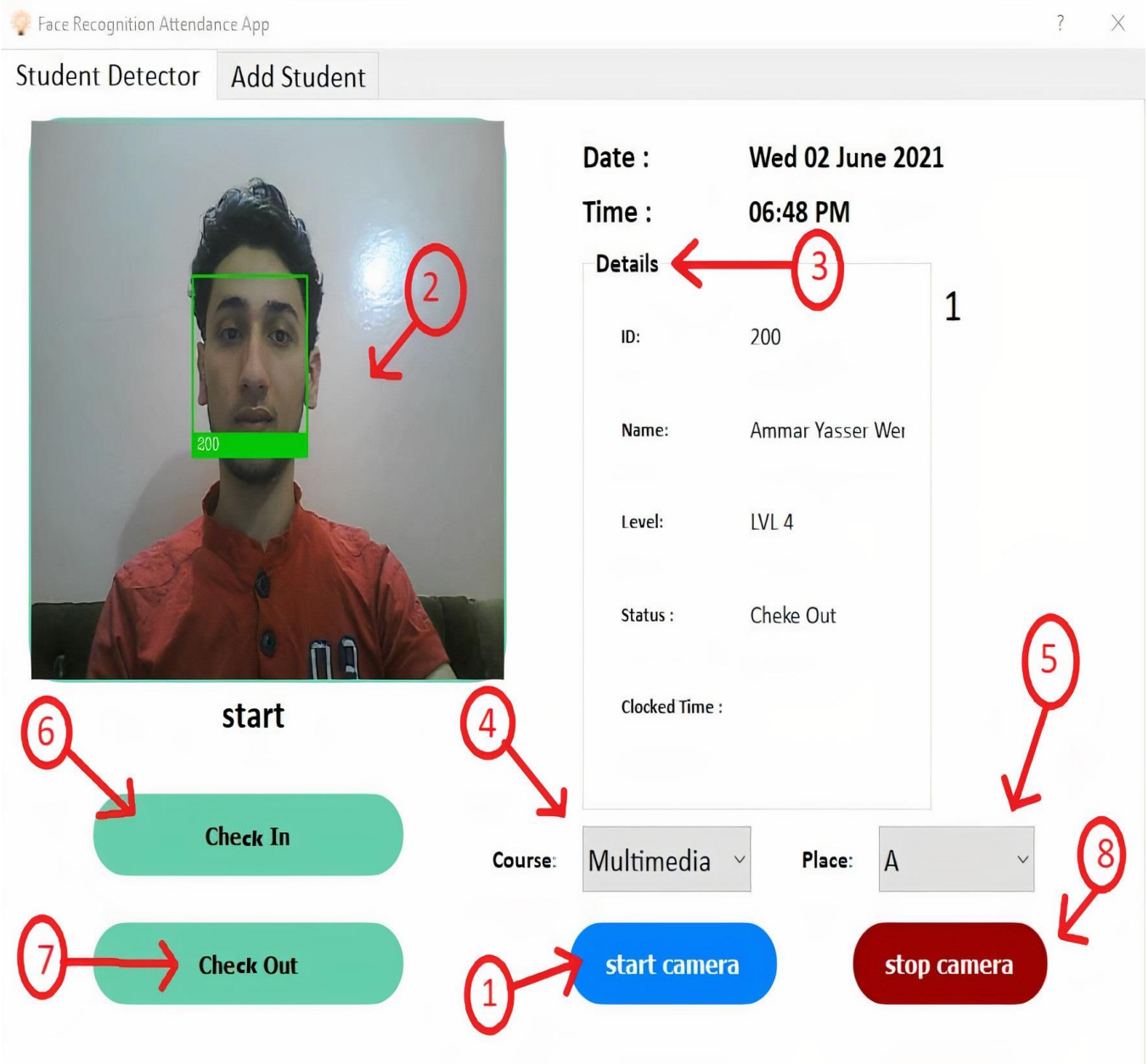

Figure 5 main form

4.10 Consisting of:

1. Start camera button - to open the camera.
2. Form to show the camera and detect face.
3. Student details form - to show details of students.
4. Drop down list for course - to choose the course.
5. Drop down list for place - to choose a place.
6. Check in button - to add all data in the csv file.
7. Check out button - to add all data in the csv file.
8. Stop camera button - to stop camera.

❖ And we added dark mode and white mode for the main form (figure 6).

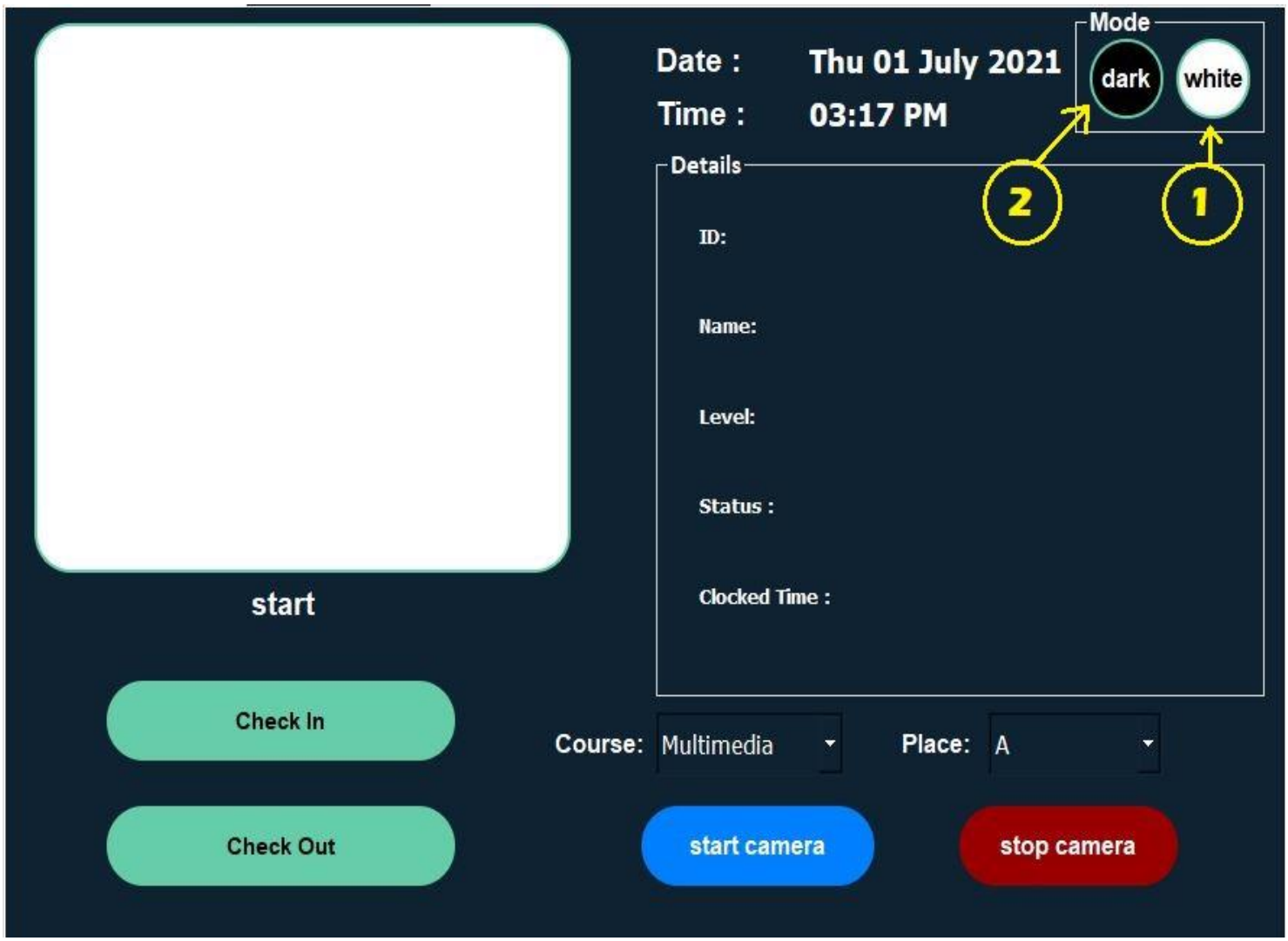

Figure 6 dark mode form.

❖Consisting of:

- ➢ White Mode button.
- ➢ Dark Mode button.

1-White mode :

If user pressed the White button, this code will be executed (figure 7).

```
def white(self):
    self.setStyleSheet("background-color: white;color: black")
    self.setWindowTitle("Color")
```

Figure 7 code for white mode.

2-dark mode :

If user pressed the White button, this code will be executed (figure 8).

```
def dark(self):
    self.setStyleSheet("background-color: #0f2231 ;color: white")
    self.setWindowTitle("Color")
```

Figure 8 code for dark mode.

4.11 Start camera button:

If user pressed the start camera button, this code will be executed (figure 9).

```python
def start(self):
        self.NameLabe.setText('start')
        self.capture = cv2.VideoCapture(int(0), cv2.CAP_DSHOW)
        self.checkInButton.setChecked(False)
        self.checkInButton.setEnabled(True)
        path = 'ImagesAttendance'
        if not os.path.exists(path):
            os.mkdir(path)
        images = []
        self.class_names = []
        self.encode_list = []
        self.TimeList1 = []
        self.TimeList2 = []
        attendance_list = os.listdir(path)
        # print(attendance_list)
        for cl in attendance_list:
            cur_img = cv2.imread(f'{path}/{cl}')
            images.append(cur_img)
            x=cl.rsplit('.', 666)
            self.class_names.append(x[0])
        for img in images:
            try:
                img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
                boxes = face_recognition.face_locations(img)
                encodes_cur_frame = face_recognition.face_encodings(img, boxes)[0]
            # encode = face_recognition.face_encodings(img)[0]
                self.encode_list.append(encodes_cur_frame)
            except Exception as e:
                print(e)
        if self.NameLabe.text()=='start':
            self.timer.timeout.connect(self.update_frame)  # Connect timeout to the output function
            self.timer.start(500)  # emit the timeout() signal at x=40ms
```

Figure 9- Start Camera.

4.12 Detect Face:-

In this point (figure 10) we will explain how the face recognition system work.

```
def face_rec_(self, frame, encode_list_known, class_names):
        def mark_attendance(idimg):
            if self.NameLabel.text()==''':
                text_files = glob.glob( "ImagesAttendance/*.jpg", recursive = True)
                print(text_files)
                print([s.strip('*/') for s in text_files]) # remove the 8 from the string
                x=[s.replace('ImagesAttendance\\', '') for s in text_files]
                print(x)
                if idimg!='unknown':
                    h=idimg
                    final_array = [i for i in range(len(x)) if h in x[i]]
                    d=final_array[0]
                    j=x[d] #name.nans.asdsad
                    print(j)
                    k=j.rsplit('.', 666)
                else:
                    self.NameLabe2.setText('0')
            if (k[1] != 'unknown'):
                self.NameLabe2.setText('1')
                self.IDLabel.setText(idimg)
                self.NameLabel.setText(k[1])
                self.lvlLabel.setText(k[2])
                with open('Attendance.csv', "r") as csv_file:
                    csv_reader = csv.reader(csv_file, delimiter=';')
                    for row in csv_reader:
                        for field in row:
                            self.StatusLabel.setText(field[0:99])
            else:
                self.NameLabe2.setText('0')
        if self.NameLabe.text()=='start':
            faces_cur_frame = face_recognition.face_locations(frame)
            encodes_cur_frame = face_recognition.face_encodings(frame, faces_cur_frame)
```

Figure 10 - Execute Code For Detect Face.

```
for encodeFace, faceLoc in zip(encodes_cur_frame, faces_cur_frame):
    match = face_recognition.compare_faces(encode_list_known, encodeFace, tolerance=0.50)
    face_dis = face_recognition.face_distance(encode_list_known, encodeFace)
    idimg = "unknown"
    best_match_index = np.argmin(face_dis)
    y1, x2, y2, x1 = faceLoc
    # print("s",best_match_index)
    if match[best_match_index]:
        idimg = class_names[best_match_index].upper()
        print('kkkkkkkkkkkkkkkkkkkkkkkkkkkk')
        y1, x2, y2, x1 = faceLoc
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 200, 0), 2)
        cv2.rectangle(frame, (x1, y2 - 20), (x2, y2), (0, 200, 0), cv2.FILLED)
        cv2.putText(frame, idimg, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)

        if  idimg!=self.IDLabel.text():
                self.NameLabe2.setText('')
                self.IDLabel.setText('')
                self.NameLabel.setText('')
                self.lvlLabel.setText('')
                self.StatusLabel.setText('')
                self.HoursLabel.setText('')
                self.MinLabel.setText('')
    mark_attendance(idimg)

return frame
```

Figure 10 - Execute Code For Detect Face.

4.13 Check in button:-

If user pressed the check in button, this code will be executed (Figure 11)

```
def ClockIn(self):
        if self.ClockInButton.isChecked():
            self.ClockInButton.setEnabled(False)
            with open('Attendance.csv', 'a') as f:
                    if (self.NameLabel.text() != '' ):
                        buttonReply = QMessageBox.question(self, 'Welcome ' + self.NameLabel.text(),
                        'Are you Clocking In?' ,QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
                        if buttonReply == QMessageBox.Yes:
                            text = str(self.comboBox.currentText())
                            text2 = str(self.comboBox2.currentText())
                            date_time_string = datetime.datetime.now().strftime("%y/%m/%d, %H:%M:%S")
                            f.writelines(f'\n{self.NameLabel.text()};{date_time_string};{text};{text2};Check In')
                            self.ClockInButton.setChecked(False)
                            self.StatusLabel.setText('Check In')
                            self.HoursLabel.setText('Measuring')
                            self.MinLabel.setText('')
                            #self.CalculateElapse(idimg)
                            #print('Yes clicked and detected')
                            self.Time1 = datetime.datetime.now()
                            #print(self.Time1)
                            self.ClockInButton.setEnabled(True)

                            self.NameLabe2.setText('')
                            self.IDLabel.setText('')
                            self.NameLabel.setText('')
                            self.lvlLabel.setText('')
                            self.StatusLabel.setText('')
                            self.HoursLabel.setText('')
```

Figure 11 - Execute Code for Check In.

```
        else:
            self.ClockInButton.setChecked(False)
            self.ClockInButton.setEnabled(True)

        time.sleep(1.5)

    else:
        buttonReply = QMessageBox.question(self, 'Welcome ' + self.NameLabe.text(),
         'You are unknown!' ,QMessageBox.Ok  )
        self.ClockInButton.setChecked(False)
        self.ClockInButton.setEnabled(True)
```

Figure 11 - Execute Code for Check In.

4.14 Check out button:-

If user pressed check out button, this code will be executed (figure 12).

```
def ClockOut(self):
        if self.ClockOutButton.isChecked():
            self.ClockOutButton.setEnabled(False)
            with open('Attendance.csv', 'a') as f:
                    if (self.NameLabel.text()!= 'unknown'):
                        buttonReply = QMessageBox.question(self, 'Cheers ' + self.NameLabel.text(),
                         'Are you Clocking Out?',QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
                        if buttonReply == QMessageBox.Yes:
                            text = str(self.comboBox.currentText())

                            text2 = str(self.comboBox2.currentText())
                            date_time_string = datetime.datetime.now().strftime("%y/%m/%d, %H:%M:%S")
                            f.writelines(f'\n{self.NameLabel.text()};{date_time_string};{text};{text2};Check Out')
                            self.ClockOutButton.setChecked(False)
                            self.StatusLabel.setText('Check Out')
                            self.Time2 = datetime.datetime.now()
                            #print(self.Time2)

                            self.ElapseList(self.NameLabel.text())
                            self.TimeList2.append(datetime.datetime.now())
                            CheckInTime = self.TimeList1[-1]
                            CheckOutTime = self.TimeList2[-1]
                            self.ElapseHours = (CheckOutTime - CheckInTime)
                            self.MinLabel.setText("{:.0f}".format(abs(self.ElapseHours.total_seconds() / 60)%60)
                            + 'm')
                            self.HoursLabel.setText("{:.0f}".format(abs(self.ElapseHours.total_seconds() / 60**2))
                            + 'h')
                            self.ClockOutButton.setEnabled(True)
                        else:
                            print('Not clicked.')
                            self.ClockOutButton.setEnabled(True)
```

Figure 12 - Execute Code For Check Out.

4.15 Stop camera button:-

If user pressed the stop camera button, this code will be executed (figure 13).

```
def stop(self):
    self.NameLabe.setText('stop')
    self.imgLabel.setText(" ")
```

Figure 13 - Execute Code For Stop Camera.

4.16 Add Student Form:-

If user pressed add student button, this form will appear (figure 14).

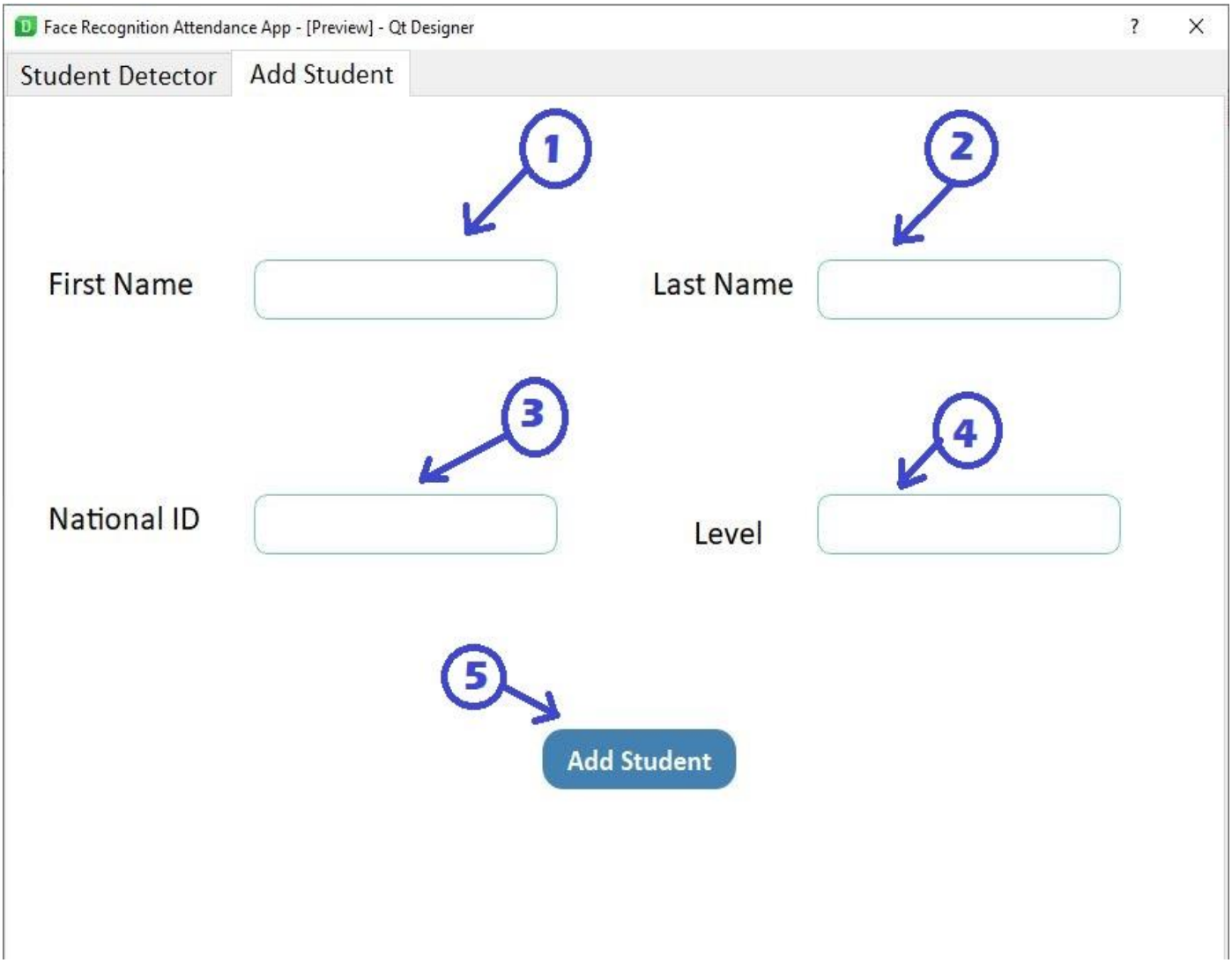

Figure 14 Add Student

4.17 Consisting of:

1- First name to add first name.

2- Last name to add last name.

3- National id to add id.

4- Level to add level.

5- Button to add data in database.

4.18 Add Student button:-

If user pressed the add student button, this code will be executed (figure 15).

```
def addstudent(self):
    try:
        con =mysql.connector.connect(user='admin',password='12345',host='localhost',database='student')
        manager = con.cursor()
        first_name= self.linefname.text()
        last_name =  self.linelname.text()
        id        =  self.lineid.text()
        level =  self.linelevel.text()
        query = "INSERT INTO students (first_name,last_name,id, level) VALUES (%s, %s, %s, %s)"
        value = (first_name,last_name,id, level)
        manager.execute(query, value)
        con.commit()
        print("Data Inserted ")
    except:
        print("Error Inserting Data")
```

Figure 15 code for add student

To view the source code visit this link:-

https://github.com/werdani/face_recognition-system