



**islington college**

(इस्लिंग्टन कॉलेज)

**Module Code & Module Title  
CS6P05NI Final Year Project**

**Assessment Weightage & Type  
40% FYP Final Report**

**Year & Semester  
2022 Spring**

**Project Title: Face Recognition Attendance System (FRAS)**

**Student Name:**

**London Met ID:**

**College ID:**

**Internal Supervisor:**

**External Supervisor:**

**Assignment Due Date: 27<sup>th</sup> April, 2022**

**Assignment Submission Date: 27<sup>th</sup> April, 2022**

**Word Count (Where required): Approx. 11,000 Words**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

## Acknowledgements

I'd would like to thank everyone who contributed to the success of this project. First and foremost, I'd want to express my gratitude to Mr. Aakash Ojha and Ms. Suruchi Shrestha, my research external and internal supervisors, who, despite their incredibly busy schedules, made the time to provide critical advice and assistance during the research's progress.

I'd also like to express my gratitude to my supporting parents and extended family for their unwavering support and encouragement. Finally, I am grateful for my friends' altruistic cooperation and support in helping me complete this task.

## Abstract

There is no reason why a fundamental educational activity like attendance should be considered in the old, boring fashion in this day and age of rapidly increasing new technology. It is challenging to manage big groups of students in a classroom using the traditional way. It is not advised since entering data into a system takes time and has a significant potential of error. Real-Time Face Recognition is a useful way for dealing with the daily attendance of a big number of pupils. Numerous algorithms and techniques can be applied to improve face recognition performance, but this proposed model employs the Haarcascade classifier to determine the positive and negative characteristics of the face, as well as the LBPH (Local binary pattern histogram) algorithm for face recognition, both of which are implemented in Python and the OpenCV library. The TKinter GUI interface is used for user interface purposes.

## TABLE OF CONTENT

<u>CHAPTER 1: INTRODUCTION .....</u>	1
<u>1. INTRODUCTION .....</u>	1
1.1 PROJECT DESCRIPTION .....	1
1.2 CURRENT SCENARIO.....	3
1.3 PROBLEM DOMAIN AND PROJECT AS A SOLUTION.....	3
1.3.1 PROBLEM DOMAIN.....	3
1.3.2 PROJECT AS A SOLUTION .....	4
1.3.3 DRAWBACK OF EXISTING SYSTEM .....	6
1.3.4 USING BIOMETRIC .....	7
1.4 AIM AND OBJECTIVES .....	8
1.4.1 AIM .....	8
1.4.2 OBJECTIVES .....	8
1.4.3 PROJECT FEATURES.....	9
1.5 CLIENT REQUIREMENTS.....	9
1.6 STRUCTURE OF THE REPORT.....	10
1.6.1 INTRODUCTION: CHAPTER 1 .....	10
1.6.2 BACKGROUND: CHAPTER 2.....	10
1.6.3 DEVELOPMENT: CHAPTER 3 .....	10
1.6.4 TESTING AND ANALYSIS: CHAPTER 4.....	11
1.6.5 CONCLUSION: CHAPTER 5.....	11
<u>CHAPTER 2: BACKGROUND .....</u>	12
<u>2. BACKGROUND .....</u>	12
2.1 ABOUT THE END USERS.....	12
2.1.1 CLIENT'S NAME AND ADDRESS .....	12
2.1.2 CLIENT APPROVAL LETTER .....	14
2.1.3 DISCUSSION ABOUT THE PROJECT WITH CLIENT .....	15
2.1.3.1 Problem.....	15
2.1.3.2 Solution.....	15
2.2 UNDERSTANDING THE SOLUTION .....	16
2.2.1 IoT AS A PLATFORM .....	16
2.2.2 PROJECT ELABORATION.....	16
2.2.3 PROJECT DELIVERIES .....	17
2.3 SYSTEM ARCHITECTURE .....	17
2.3.1 SYSTEM ARCHITECTURE OF FACE RECOGNITION ATTENDANCE SYSTEM .....	17
2.3.2 PROJECT OVERVIEW ARCHITECTURE .....	18
2.4 TOOLS AND TECHNOLOGIES .....	18
2.4.1 XAMPP (MYSQL SERVER): DATABASE .....	18
2.4.2 IDE: PYCHARM.....	19

2.4.3	PYTHON PROGRAMMING LANGUAGE .....	19
2.4.4	GUI FRAMEWORK: TKINTER .....	20
2.4.5	PYTHON LIBRARIES .....	21
2.4.6	CLASSIFIER AND ALGORITHMS.....	21
2.4.6.1	Face Recognition Using HAAR-Cascade.....	21
2.4.6.2	Local Binary Pattern Histogram Face Recognizer (LPBH) Classifier .....	23
2.5	SIMILAR PROJECTS .....	25
2.5.1	PROJECT 1: SMART STUDENT ATTENDANCE SYSTEM USING QR CODE .....	25
2.5.2	PROJECT 2: SMART ATTENDANCE MONITORING SYSTEM USING IOT AND RFID .....	27
2.5.3	PROJECT 3: SMART ATTENDANCE SYSTEM USING FINGER PRINT BIOMETRIC .....	28
2.6	COMPARISONS .....	29
2.6.1	COMPARISON TABLE .....	29
2.6.2	CONCLUSION FROM THE SIMILAR PROJECTS.....	30
	<b>CHAPTER 3: DEVELOPMENT.....</b>	<b>31</b>
<b>3.</b>	<b><u>DEVELOPMENT</u></b> .....	<b>31</b>
3.1	CONSIDERED METHODOLOGY .....	31
3.2	SELECTED METHODOLOGY: EVOLUTIONARY PROTOTYPING.....	31
3.3	COMPARISON BETWEEN ANY TWO METHODOLOGIES .....	33
3.4	SIMILARITY BETWEEN EVOLUTIONARY PROTOTYPE MODEL AND ITERATIVE WATERFALL MODEL 34	
3.5	PHASES OF METHODOLOGY .....	34
3.5.1	PHASES OF PROTOTYPE.....	34
3.5.1.1	Requirement Gathering and Analysis .....	34
3.5.1.2	Quick Design .....	34
3.5.1.3	Build prototype .....	35
3.5.1.4	Initial user evaluation .....	35
3.5.1.5	Refining prototype .....	35
3.5.1.6	Implement and maintain .....	35
3.6	SYSTEM REQUIREMENTS ANALYSIS .....	36
3.6.1	TECHNICAL REQUIREMENT.....	36
3.6.1.1	Hardware Requirement.....	36
3.6.1.2	Software Requirement .....	36
3.6.2	FUNCTIONAL REQUIREMENT .....	36
3.6.3	NON-FUNCTIONAL REQUIREMENT .....	37
3.6.4	ADMIN/TEACHER REQUIREMENT .....	37
3.7	DESIGN.....	38
3.7.1	USE CASE DIAGRAM .....	38
3.7.2	UML SEQUENCE DIAGRAM .....	39
3.7.3	FLOW CHART .....	40
3.7.3.1	Admin Flowchart .....	40
3.7.3.2	Attendance Flowchart .....	41
3.7.4	ENTITY RELATIONAL DIAGRAM .....	42
3.8	IMPLEMENTATION OF METHODOLOGY .....	43
3.8.1	ITERATION: .....	43

3.8.1.1	Requirement Gathering.....	43
3.8.1.2	Quick Design .....	43
3.8.1.3	Build a prototype .....	44
3.8.1.4	User Evaluation: .....	53
3.8.1.5	Refining a prototype:.....	53
3.8.1.6	Implement the features .....	53
3.8.2	GUI DESIGN.....	54
3.8.2.1	Login Window .....	54
3.8.2.2	Register Window .....	55
3.8.2.3	Forgot Password Window.....	56
3.8.2.4	Home Page Window .....	58
3.8.2.5	Student Detail Window .....	60
3.8.2.6	Train Data Window .....	62
3.8.2.7	Face Recognition Window .....	64
3.8.2.8	Attendance Window .....	66
3.8.2.9	Developer Window.....	68
3.8.2.10	Help Desk Window .....	70

<b>CHAPTER 4: TESTING AND ANALYSIS .....</b>	<b>72</b>
--	-----------

<b>4. TESTING AND ANALYSIS .....</b>	<b>72</b>
--------------------------------------	-----------

4.1	TEST PLAN .....	72
4.1.1	TEST PLAN FOR WHITE BOX TESTING .....	72
4.1.2	TEST PLAN FOR BLACK BOX TESTING .....	72
4.1.3	TEST PLAN FOR INTEGRATION TESTING .....	73
4.2	WHITE BOX TESTING.....	74
4.2.1	SUCCESSFUL TEST CASE FOR ADMIN .....	74
4.2.1.1	Test 1 – To display the login page.....	74
4.2.1.2	Test 02: To login to the system.....	75
4.2.1.3	Test 03: To Register New Teacher .....	77
4.2.1.4	Test 04: To reset password .....	80
4.2.1.5	Test 05: To login new registered teacher.....	82
4.2.1.6	Test 06: Add New Student detail.....	84
4.2.1.7	Test 07: To update information of existing student .....	87
4.2.1.8	Test 08: To reset filled-up information in text areas .....	89
4.2.1.9	Test 09: To delete the existing student details.....	90
4.2.1.10	Test 10: To Train all dataset in database .....	92
4.2.1.11	Test 11: To start face recognition from ‘Face Recognition’ window .....	94
4.2.1.12	Test 12: To verify attendance marked in csv file.....	96
4.2.1.13	Test 13: To import csv file into ‘attendance’ window .....	97
4.2.1.14	Test 14: To modify CSV attendance file and export .....	98
4.2.1.15	Test 15: To reset all text field on single click.....	100
4.3	BLACK BOX TESTING/FAILURE TEST CASES.....	102
4.3.1	ERROR 01- IN TEACHER REGISTRATION, STRONG PASSWORD IS NOT RECOMMENDED .....	102
4.3.2	ERROR 02- DATA FETCHED FORM DATABASE WHILE FACE RECOGNITION IS INCLUDED WITH TUPLES .....	105

4.3.3 ERROR 03- UNABLE TO LAUNCH CAMERA.....	107
4.4 INTEGRATION TESTING .....	111
4.4.1 INTEGRATION OF DATABASE WITH THE APPLICATION.....	111
4.4.1.1 Available Students.....	111
4.4.2 CRUD ON DATABASE.....	111
4.4.2.1 Test 16: To edit a contact number of student in database.....	111
4.4.2.2 Test 17: To delete one of the students .....	113
4.4.2.3 Test 18: To edit security answer in database .....	114
4.5 CRITICAL ANALYSIS .....	116
 <u>CHAPTER 5: CONCLUSION.....</u>	 117
 <u>5. CONCLUSION .....</u>	 117
5.1 LEGAL, SOCIAL AND ETHICAL ISSUES .....	117
5.1.1 LEGAL ISSUES .....	118
5.1.2 SOCIAL ISSUES .....	118
5.1.3 ETHICAL ISSUES.....	118
5.2 ADVANTAGES .....	119
5.2.1 AUTOMATED TIME TRACKING SYSTEM.....	119
5.2.2 COST-EFFECTIVE.....	119
5.2.3 TIME SAVING.....	119
5.2.4 INCREASED SECURITY .....	120
5.2.5 EASY TO MANAGE.....	120
5.3 SCOPE AND LIMITATIONS.....	121
5.3.1 SCOPE OF PROJECT .....	121
5.3.2 LIMITATION OF PROJECT .....	121
5.4 FUTURE WORKS.....	122
 <u>CHAPTER 6: REFERENCE .....</u>	 123
 <u>6. REFERENCE .....</u>	 123
 <u>CHAPTER 7: BIBLIOGRAPHY .....</u>	 126
 <u>CHAPTER 8: APPENDIX .....</u>	 130
 <u>7. APPENDIX .....</u>	 130
8.1 CONSIDERED METHODOLOGY .....	130
8.1.1 AGILE DEVELOPMENT METHODOLOGY.....	130
8.1.2 WATERFALL DEVELOPMENT METHODOLOGY .....	131
Pros .....	131
Cons .....	131
8.1.3 PROTOTYPE MODEL.....	132

Pros .....	132
Cons .....	132
8.2 APPENDIX B: PYTHON LIBRARIES .....	133
8.2.1 PILLOW .....	133
8.2.2 MYSQL CONNECTOR .....	133
8.2.3 OPENCV.....	134
8.2.4 OS.....	134
8.2.5 CSV .....	134
8.2.6 DATETIME.....	135
8.2.7 TIME.....	135
8.2.8 TKCALENDAR .....	135
8.3 APPENDIX C: FUNCTION AND FEATURE .....	136
8.3.1 REGISTER .....	136
8.3.2 LOGIN.....	136
8.3.3 FORGET PASSWORD .....	136
8.3.4 MODIFY STUDENT DETAIL.....	136
8.3.5 TAKE STUDENT PHOTO.....	136
8.3.6 GENERATE DATASET.....	136
8.3.7 TRAIN DATASET .....	136
8.3.8 MODIFY ATTENDANCE.....	136
8.3.9 IMPORT/EXPORT ATTENDANCE SHEET .....	136
8.4 APPENDIX C: PRE-SURVEY .....	137
8.4.1 PRE-SURVEY FORM.....	137
8.4.2 SAMPLE OF FILLED PRE-SURVEY FORMS .....	141
8.4.3 PRE-SURVEY RESULT .....	153
8.5 APPENDIX B: POST-SURVEY .....	154
8.5.1 POST-SURVEY FORM.....	154
8.5.2 SAMPLE OF FILLED POST-SURVEY FORMS.....	157
8.5.3 POST-SURVEY RESULT .....	165
8.6 APPENDIX C: SAMPLE CODES .....	166
8.6.1 SAMPLE CODE OF THE UI.....	166
8.6.1.1 Login Window .....	166
8.6.1.2 Register window .....	169
8.6.1.3 Student Window .....	172
8.6.1.4 Train Window.....	178
8.6.1.5 Recognition Window .....	180
8.6.1.6 Attendance Window .....	182
8.6.1.7 Help Desk window .....	187
8.6.1.8 Developer Window .....	189
8.6.2 SAMPLE CODE FOR THE AUTOMATION SCRIPT .....	191
8.6.2.1 Login.....	191
8.6.2.2 Register Window .....	193
8.6.2.3 Student Window .....	194
8.6.2.4 Train Window.....	201
8.6.2.5 Recognition Window .....	202
8.6.2.6 Attendance Window .....	205
8.7 APPENDIX D: DESIGNS.....	207
8.7.1 GANTT CHART .....	207

8.7.1.1 Initial Gantt Chart .....	207
8.7.1.2 Final Gantt Chart .....	208
8.7.2 WORK BREAKDOWN STRUCTURE.....	208
8.7.3 ALGORITHMS .....	209
8.7.3.1 Admin/Teacher login Algorithm .....	209
8.7.3.2 Recognition and attendance algorithm .....	209
8.7.4 ACTIVITY DIAGRAM .....	210
8.7.5 DATA FLOW DIAGRAM .....	211

## Table of figure

Figure 1: Face Recognition Attendance System overview .....	2
Figure 2: Annual Facial Biometric revenue by region (Martin, 2015) .....	6
Figure 3: Client's Logo .....	12
Figure 4: Client Meeting with feedback.....	13
Figure 5: Client Approval Letter.....	14
Figure 6: System Architecture .....	18
Figure 7: XAMPP Logo .....	19
Figure 8: PyCharm Logo .....	19
Figure 9: Face Feature .....	21
Figure 10: Face Feature ii .....	22
Figure 11: HAAR filter Overlay on Face .....	22
Figure 12: Sample 3x3 Grid of a Recognizer.....	23
Figure 13: Output of LBPH from original image .....	24
Figure 14: Histogram Representation of LBPH Result .....	24
Figure 15: Effect on Lighting on LBPH Faces.....	25
Figure 16: Smart Student Attendance System Using QR Code (Elbehiery, 2019).....	26
Figure 17: Smart Attendance Monitoring System Using IoT And RFID (taifur, 2018).....	28
Figure 18: Evolutionary Prototype .....	31
Figure 19: USE Case Diagram .....	38
Figure 20: UML Sequence Diagram .....	39
Figure 21: Admin Flowchart .....	40
Figure 22: Attendance Flowchart.....	41
Figure 23: Entity Relational Diagram.....	42
Figure 24: Wireframe - Login Window .....	44
Figure 25: Wireframe – Login Window.....	45
Figure 26: Wireframe – Forgot Password .....	45
Figure 27: Wireframe – Home Window .....	46
Figure 28: Wireframe - Student Window .....	47
Figure 29: Wireframe - Train window .....	48
Figure 30: Wireframe - Face Recognition.....	49
Figure 31: Wireframe - Attendance Window.....	50
Figure 32: Wireframe - Developer window.....	51
Figure 33: Wireframe - Help Desk Window.....	52
Figure 34: Code - Login Window .....	54
Figure 35: GUI - Login window .....	55
Figure 36: Code - Register window i.....	55
Figure 37: GUI - Register window .....	56
Figure 38: Code - Register window.....	57
Figure 39: Code - Register window .....	57
Figure 40: Code - Home window (i) .....	58
Figure 41: Code - Home Window ii .....	59

Figure 42: GUI - Home window .....	59
Figure 43: Code - Student detail window i .....	60
Figure 44: Code - Student detail ii .....	60
Figure 45: GUI - Student detail .....	61
Figure 46: Code - Train data .....	62
Figure 47: GUI - Train data .....	63
Figure 48: Code - Face recognition window i .....	64
Figure 49: Code - Face recognition window ii .....	65
Figure 50: Code - Face Recognition window .....	65
Figure 51: Code - Attendance window .....	66
Figure 52: GUI - Attendance window .....	67
Figure 53: Code - Developer window .....	68
Figure 54: GUI - Developer window .....	69
Figure 55: Code - Help Desk window .....	70
Figure 56: GUI - Help desk window .....	71
Figure 57: Test 1 (i) .....	74
Figure 58: Test 1 (ii) .....	75
Figure 59: Test 2 (i) .....	76
Figure 60: Test 2 (ii) .....	76
Figure 61: Test 2 (iii) .....	77
Figure 62: Test 03 (i) .....	78
Figure 63: Test 03 (ii) .....	78
Figure 64: Test 03 (iii) .....	79
Figure 65: Test 03 (iv) .....	79
Figure 66: Test 04 (i) .....	80
Figure 67: Test 04 (ii) .....	81
Figure 68: Test 04 (iii) .....	81
Figure 69: Test 04 (iv) .....	82
Figure 70: Test 05 (i) .....	83
Figure 71: Test 05 (ii) .....	83
Figure 72: Test 05 (iii) .....	84
Figure 73: Test 06 (i) .....	85
Figure 74: test 06 (ii) .....	85
Figure 75: Test 06 (iii) .....	86
Figure 76: Test 06 (iv) .....	86
Figure 77: Test 07 (i) .....	87
Figure 78: Test 07 (ii) .....	88
Figure 79: Test 07 (iii) .....	88
Figure 80: Test 08 (i) .....	89
Figure 81: Test 08 (ii) .....	90
Figure 82: Test 09 (i) .....	91
Figure 83: Test 09 (ii) .....	91
Figure 84: Test 09 (iii) .....	92

Figure 85: Test 10 (i).....	93
Figure 86: Test 10 (ii).....	93
Figure 87: Test 10 (iii).....	94
Figure 88: Test 11 (i).....	95
Figure 89: Test 11 (ii).....	95
Figure 90: Test 12 (i).....	96
Figure 91: Test 12 (ii).....	97
Figure 92: Test 13 (i).....	97
Figure 93: Test 13 (ii).....	98
Figure 94: Test 14 (i).....	99
Figure 95: Test 14 (ii).....	99
Figure 96: Test 14 (iii).....	100
Figure 97: Test 15 (i).....	101
Figure 98: Test 15 (ii).....	101
Figure 99: Error 01 (i) .....	102
Figure 100: Error 01 (ii) .....	103
Figure 101: Error 01 (iii) .....	103
Figure 102: Error 01 (iv).....	104
Figure 103: Error 01 (v).....	104
Figure 104: Error 02 (i) .....	105
Figure 105: Error 02 (ii) .....	105
Figure 106: Error 02 (iii) .....	106
Figure 107: Error o2 (iv).....	107
Figure 108: Error 03 (i) .....	108
Figure 109: Error 03 (ii) .....	108
Figure 110: Error 03 (iii) .....	109
Figure 111: Error 03 (iv).....	109
Figure 112: Error 03 (v).....	110
Figure 113: Available students (i) .....	111
Figure 114: Available students (ii) .....	111
Figure 115: Test 16 (i).....	112
Figure 116: Test 16 (ii).....	112
Figure 117: Test 16 (iii).....	112
Figure 118: Test 16 (iv) .....	113
Figure 119: Test 17 (i).....	113
Figure 120: Test 17 (ii).....	114
Figure 121: Test 17 (iii).....	114
Figure 122: Test 18 (i).....	115
Figure 123: Test 18 (ii).....	115
Figure 124: Test 18 (iii).....	115
Figure 125: Test 18 (iv) .....	115
Figure 126: Logo of Pillow Library .....	133
Figure 127: Pre-Survey form (i) .....	137

Figure 128: Pre-Survey form (ii) .....	138
Figure 129: Pre-Survey form (iii) .....	139
Figure 130: Pre-Survey form (iv).....	140
Figure 131: Pre-Survey Response (i).....	141
Figure 132: Pre-Survey Response (ii).....	142
Figure 133: Pre-Survey Response (iii).....	142
Figure 134: Pre-Survey Response (iv).....	143
Figure 135: Pre-Survey Response (v).....	144
Figure 136: Pre-Survey Response (vi).....	144
Figure 137: Pre-Survey Response (vi).....	145
Figure 138: Pre-Survey Response (vii).....	145
Figure 139: Pre-Survey Response (viii).....	146
Figure 140: Pre-Survey Response (ix).....	146
Figure 141: Pre-Survey Response (x).....	147
Figure 142: Pre-Survey Response (xi).....	147
Figure 143: Pre-Survey Response (xii).....	148
Figure 144: Pre-Survey Response (xiii).....	148
Figure 145: Pre-Survey Response (xiv).....	149
Figure 146: Pre-Survey Response (xv) .....	149
Figure 147: Pre-Survey Response (xvi) .....	150
Figure 148: Pre-Survey Response (xvii) .....	151
Figure 149: Pre-Survey Response (xviii) .....	152
Figure 150: Post-Survey Form (i) .....	154
Figure 151: Post-Survey Form (ii) .....	155
Figure 152: Post-Survey Form (iii) .....	156
Figure 153: Post-Survey Form (iv) .....	157
Figure 154: Post-Survey Responses (i) .....	157
Figure 155: Post-Survey Responses (ii) .....	158
Figure 156: Post-Survey Responses (iii) .....	158
Figure 157: Post-Survey Responses (iv) .....	158
Figure 158: Post-Survey Responses (v) .....	159
Figure 159: Post-Survey Responses (vi) .....	159
Figure 160: Post-Survey Responses (vii) .....	160
Figure 161: Post-Survey Responses (viii) .....	160
Figure 162: Post-Survey Responses (ix) .....	161
Figure 163: Post-Survey Responses (x) .....	161
Figure 164: Post-Survey Responses (xi) .....	162
Figure 165: Post-Survey Responses (xii) .....	162
Figure 166: Post-Survey Responses (xiii) .....	163
Figure 167: Post-Survey Responses (xiv) .....	163
Figure 168: Post-Survey Responses (xv) .....	164
Figure 169: Post-Survey Responses (xvi) .....	164
Figure 170: Login Window (i).....	166

Figure 171: Login Window (ii).....	166
Figure 172: Login Window (iii).....	167
Figure 173: Login Window (iv).....	167
Figure 174: Login Window (v).....	168
Figure 175: Register window (i) .....	169
Figure 176: Register window (ii) .....	170
Figure 177: Register window (iii) .....	171
Figure 178: Register window (iv) .....	171
Figure 179: Register window (v) .....	172
Figure 180: Register window (vi) .....	173
Figure 181: Register window (vii) .....	174
Figure 182: Register window (viii) .....	174
Figure 183: Register window (ix) .....	175
Figure 184: Register window (x) .....	175
Figure 185: Register window (xi) .....	176
Figure 186: Register window (xii) .....	176
Figure 187: Register window (xiii) .....	177
Figure 188: Register window (xiv) .....	177
Figure 189: Train window (xv).....	178
Figure 190: Train window (xvi).....	179
Figure 191: Recognition window (i) .....	180
Figure 192: Recognition window (ii).....	181
Figure 193: Attendance window (i).....	182
Figure 194: Attendance window (ii).....	183
Figure 195: Attendance window (iii).....	184
Figure 196: Attendance window (iv).....	185
Figure 197: Attendance window (v).....	185
Figure 198: Attendance window (vi).....	186
Figure 199: Help Desk Window (i) .....	187
Figure 200: Help Desk Window (ii) .....	188
Figure 201: Developer window (i) .....	189
Figure 202:Developer window (ii) .....	190
Figure 203: Login Window (i).....	191
Figure 204: Login Window (ii).....	192
Figure 205: Login Window (iii).....	192
Figure 206: Register window (i) .....	193
Figure 207: Student window (i) .....	194
Figure 208: Student window (ii) .....	195
Figure 209: Student window (iii) .....	196
Figure 210: Student window (iv) .....	197
Figure 211: Student window (v) .....	198
Figure 212: Student window (vi) .....	198
Figure 213: Student window (vii) .....	199

Figure 214: Student window (viii) .....	200
Figure 215: Train window.....	201
Figure 216: Recognition window (i) .....	202
Figure 217: Recognition window (ii).....	203
Figure 218: Recognition window (iii).....	204
Figure 219: Attendance window (i).....	205
Figure 220: Attendance window(ii).....	205
Figure 221: Attendance window (iii).....	206
Figure 222: Initial Gantt Chart .....	207
Figure 223: Final Gantt Chart.....	208
Figure 224: Work Breakdown Structure .....	208
Figure 225: Activity diagram .....	210
Figure 226: Data Flow Diagram - Context Level .....	211
Figure 227: Data flow diagram - Admin .....	211
Figure 228: Data Flow Diagram – Teacher .....	212

## Table of Tables

Table 1: Similar project comparison table.....	29
Table 2: Comparison between any two methodologies.....	33
Table 3: Similarity between evolutionary and iterative waterfall model .....	34
Table 4: Test 1- To display the login page .....	74
Table 5:Test 02: To login to the system .....	75
Table 6: Test 03: To Register New Teacher .....	77
Table 7: Test 04: To reset password.....	80
Table 8: Test 05: To login new registered teacher.....	82
Table 9: Test 06: Add New Student detail .....	84
Table 10: Test 07: To update information of existing student.....	87
Table 11: Test 08: To reset filled-up information in text areas .....	89
Table 12: Test 09: To delete the existing student details .....	90
Table 13: Test 10: To Train all dataset in database .....	92
Table 14: Test 11: To start face recognition from ‘Face Recognition’ window .....	94
Table 15: Test 12: To verify attendance marked in csv file .....	96
Table 16: Test 13: To import csv file into ‘attendance’ window .....	97
Table 17: Test 14: To modify CSV attendance file and export .....	98
Table 18: Test 15: To reset all text field on single click.....	100
Table 19: Test 16: To edit a contact number of student in database .....	111
Table 20: Test 17: To delete one of the students .....	113
Table 21: Test 18: To edit security answer in database.....	114

## List of abbreviation

AFR	Automated face recognition
CNN	Convolution Neural Network
GUI	Graphical User Interface
LBPH	Local Binary Pattern Histogram
MYSQL	My Structured Query Language
RAM	Random Access Memory
ReLU	Rectified Linear Units
RFID	Radio frequency identification
SQL	Structured Query Language
OpenCV	Open Computer vision
CSV	Comma-Separated Values

## Key Words

Face Detection  
 Face Recognition  
 CSV  
 Python  
 OpenCV  
 Tkinter GUI  
 Haarcascade Classifier  
 Local Binary Pattern Histogram (LBPH)

# Face Recognition Attendance System (FRAS)

## Chapter 1: Introduction

### 1. Introduction

#### 1.1 Project Description

These days, technology attempts to convey a significant amount of knowledge-based technical innovation. Teachers used to manually take attendance in the classroom at the beginning and end of each class period. The problem with this strategy is that it takes time to take attendance, and the manual process is prone to errors in the majority of cases. RFID (Radio Frequency Identification) was established in recent years to address this issue. Those, however, have a fail-proof of-attendance system. Some organizations choose a document-oriented strategy, while others have adopted digital tactics such as biometric fingerprinting and card swapping. These solutions, however, prove to be a statute of limitations because they require pupils to wait in a time-consuming line. The student will not be able to acquire attendance if he does not present his ID card. In a changing world, evolving technology have achieved numerous advancements (Patnaik, 2020). Other biometric approaches can also be utilized to computerize the attendance process, such as those listed below:

- Log Book entry
- Fingerprint based System
- IRIS Recognition
- RFID based System
- Face Recognition

As a result, I'd like to introduce the Face Recognition Based Student Attendance System concept. The suggested system's major goal is to assign attendance to students using face recognition-based algorithms in order to establish a fail-proof attendance system. Face detection is a technique for identifying human faces in digital photos or video that is used in a variety of applications. It is described as a subset of object-class detection in which the locations and sizes of all objects in an image that belong to a certain class are determined. Regardless of orientation, lighting circumstances, or skin color, the system can anticipate frontal or near-frontal faces in a picture.

Face Recognition is a type of biometric software that uses mathematics to map an individual's facial traits and saves the information as a face print. To validate an individual's identification, the software uses Deep Learning algorithms to match a live capture or digital image to a saved face print.

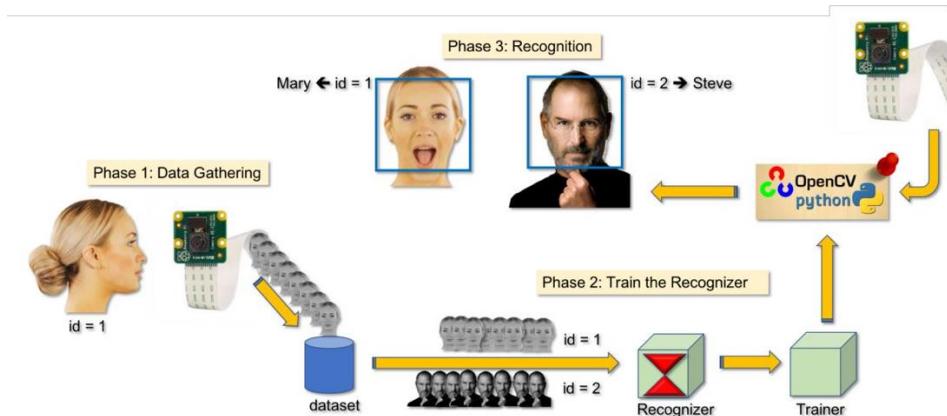


Figure 1: Face Recognition Attendance System overview

## 1.2 Current Scenario

With 30,000 contract employees in Southeast Asia, a multinational firm contacted Chennai-based Ramco Systems, which offers Human Capital Management (HCM) solutions for a variety of essential industries, with an unusual challenge on its shop floor. It said that while it takes an average of 15 minutes for a worker to reach his machine from the moment he enters the facility, employees in one of its units were taking 45 minutes. This meant that every day, the corporation was wasting 30 minutes of man-hours per employee (Narasimhan, 2019).

Several projects are competing for first place in the biometric innovation contest. Google, Apple, Facebook, Amazon, and Microsoft (GAFAM) are all present and correct. To advance our understanding as quickly as possible, all of the software online giants now disclose their theoretical breakthroughs in artificial intelligence, picture recognition, and face analysis on a regular basis. Researchers at The Chinese University of Hong Kong created the GaussianFace algorithm in 2014, which obtained face recognition scores of 98.52 percent, compared to 97.53 percent for people. Despite flaws in memory capacity and calculating times, this is a good rating (Thales, 2021).

## 1.3 Problem Domain and Project as a Solution

### 1.3.1 Problem Domain

It has been a manual system since the beginning, which is inherently flawed due to the numerous issues that have arisen as a result of its implementation. The following are some of the primary issues with traditional attendance: -

- Risk of human error - When using traditional manual attendance, it can lead to spelling errors and other payroll difficulties.

- Time consuming – Taking attendance, calling out names, and completing signatures takes a long time, which interferes with working or studying time.
- Ineffective and outdated – Although a small number of attendees may be feasible, thousands of attendees increase the rate of error, which may have an impact on the company's performance.
- Keyboard and printing error – Thousands of employees' attendance is plagued by keyboard typing and printing errors due to human nature's flaws.
- Incorrect entry of times – Time theft methods such as punching a friend might impair traditional attendance, putting a strain on the company's finances.
- Too much paperwork – Paperwork can take up a lot of space if attendance is documented in each batch's record. It complicates the retrieval of specific employee data or batch data, suggesting that files are more difficult to get.

### 1.3.2 Project as a Solution

Based on the presented problem of manual attendance, Face recognition is the ability of a computer system to detect and recognize human faces in photos or videos quickly and accurately. For enhancing the performance of face recognition, a variety of algorithms and strategies have been developed. Face recognition has been an important component of biometrics in recent years. Basic human characteristics are linked to previously collected data in biometrics. Face-detection and recognition computers could be used for a variety of purposes, including criminal identification, security systems, and identity verification. This project aims solving problems which are enlisted below: -

- Time efficient – It solves the problem of calling out the name of student taking attendance within a second.
- Risk of human error – Human involvement is reduced because this is an automated system.
- Most relevant than other biometric system – It is evolving which leads to most adopted system collaborating with finger print too.
- No keyboard or print required – As being automated system, almost no keyboard is required for attendance.
- Exact entry time record – This system being automated, entry time is auto detected and recorded in database.
- No paperwork required – This computerized system require almost no any paperwork at all.

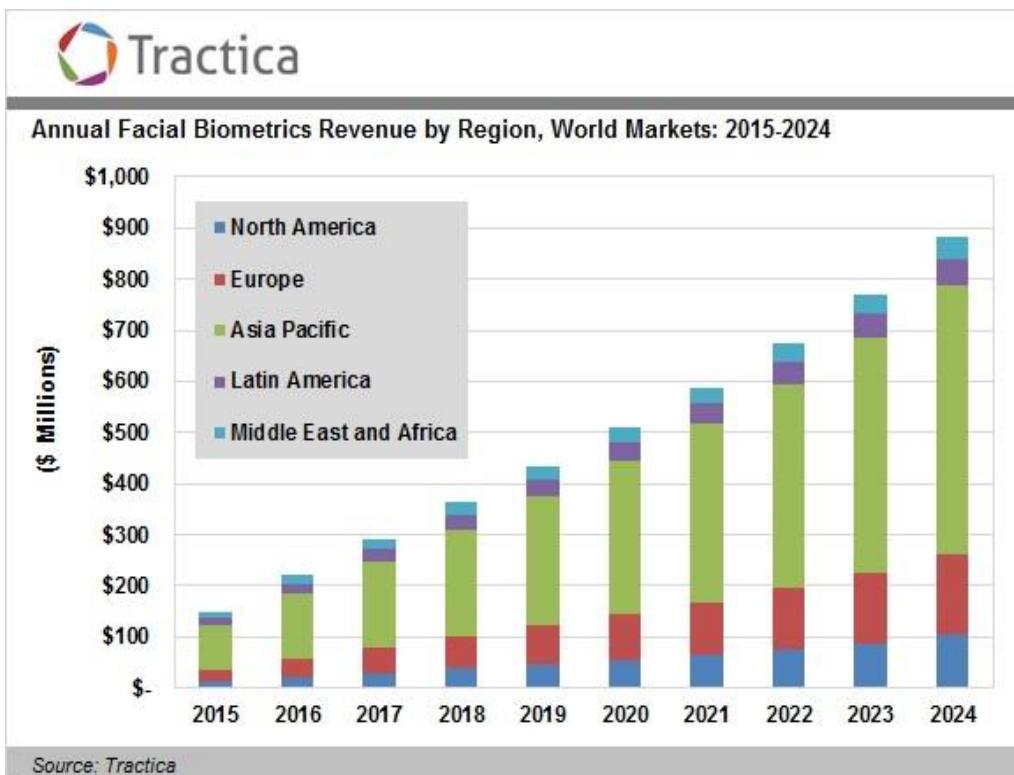


Figure 2: Annual Facial Biometric revenue by region (Martin, 2015)

A graph of expected yearly facial biometric revenue by region from 2015 to 2024 was shown in a study. This means that facial biometrics are in high demand all around the world. It is estimated to generate roughly 900 million dollars in revenue until 2024.

### 1.3.3 Drawback of Existing System

Manual systems put pressure on people to be excellent in all aspects of their work at all times, which is problematic because people aren't perfect. However, some of the system's disadvantages are outlined below:

- These are manual attendance systems.
- Forgery (one person signing in the presence of another) is always a possibility because everything is done manually, therefore there is a high danger of error and fraud.
- More personnel are necessary.

- Attendance computations (total classes attended in a month) are done by hand, which is prone to error.
- Maintaining a database or registering in manual processes is tough.
- Finding a specific piece of information in this system (particularly. If the information is from a long time ago).
- Impersonation could be easy using this system, and the attendance sheet could be stolen or lost.

#### **1.3.4 Using Biometric**

Biometrics is a new technology that uses biological characteristics such as fingerprints, iris scans, and facial recognition to automatically identify a person. A common biometric technology is the fingerprint verification mechanism (Ronak Dalvi, Apr 2019). Companies have traditionally used a thumb-based attendance system as a good substitute for card-based systems. Neither of them was ideal, and they both had dependability and security issues. Face recognition attendance has gotten a lot of attention in recent years. While many large corporations have upgraded to face recognition attendance systems, many mid-sized businesses are still using older technology.

## 1.4 Aim and Objectives

### 1.4.1 Aim

This project's main purpose is to assist lecturers in improving and organizing the process of tracking and managing student/employee attendance and absenteeism.

### 1.4.2 Objectives

The proposed system will reduce the paperwork where attendance will no longer involve any manual recording. The new system will also reduce the total time needed to do attendance recording. The new system will acquire individual attendance by means of facial recognition to secure data accuracy of the attendance.

The objectives of this project are: -

- To learn the facial recognition system and AI technology.
- To guarantee that the attendance recording procedure is speedier than the previous system, which could record each student's attendance in as little as 3-5 seconds.
- To detect unique faces with the help of computer's camera
- Create a database for the attendance tracking software.
- Based on the face database, be able to reliably distinguish an individual's face as well as several faces.
- Make it easy for administrators to access the attendance database by providing a user-friendly interface.
- Using a GUI, allow new students to register or store their faces in the database.
- Capturing live feed from camera to record attendance time.
- Detect face even in a video or image files.

#### 1.4.3 Project features

The project was a set of interconnected tasks that were to be completed over a specific period of time with the goal of solving a modern problem with a modern solution. It is generally considered a success if it meets the objectives within the agreed-upon timeframe and according to the acceptance criteria. The following are some of the characteristics featured in this diverse project: -

- Long term storage of records
- High Accuracy in calculation
- Time saving
- Optimize the resources
- Efficiency in modification, sorting and retrieval of data
- Inexpensive updating in facilities and terms of organizations.

#### 1.5 Client Requirements

- **Automate the attendance system in systematic format.**

As per requirement, the system should be designed which is able to detect face of specified person and store name and time in database.

- **System having GUI interface**

The system should have GUI as frontend which must be accessed by admin. Admin login panel should be user-friendly for easy to use.

- **Attendance data stored in database**

After the face recognition followed by attendance mark, the system should be able to store attendance data in separate file. Admin should have access to view and modify attendance database.

- **Enrol any people whenever admin desire.**

System should be designed in such way that through admin panel, admin should be able to enroll or remove any people whenever he/she desire.

- **Display name and ID of recognized face on live video.**

The system should be able to display name and ID of recognized person on live video with high accuracy.

- **Permitting admin to modify the student absent or late.**

Admin should have access to modify the attendance of absent or late students.

## 1.6 Structure of the Report

### 1.6.1 Introduction: Chapter 1

The project is introduced in the first chapter, along with its current scenarios. In addition, the problem domain and project that is a solution to the current challenge are briefly described. This chapter lists the project's goals and objectives, as well as its scope and limitations.

### 1.6.2 Background: Chapter 2

The project's requirements and target market are clarified in the backdrop, providing for a better understanding of the project. It also offers you a better sense of similar projects and the technical aspects of the program.

### 1.6.3 Development: Chapter 3

The development of the project is discussed in the development chapter. It explores the different steps of the chosen approach and outlines the considerations to consider while picking a methodology.

**1.6.4 Testing and Analysis: Chapter 4**

This chapter covers the approaches for white box testing, black box testing, integration testing, and customer evaluation testing. It also contains information about the best solution for the project, evaluation, and system operations.

**1.6.5 Conclusion: Chapter 5**

This chapter discusses the report's premise, an evaluation of the project's broader ramifications, and future improvements.

## Chapter 2: Background

### 2. Background

#### 2.1 About the End Users

##### 2.1.1 Client's Name and Address

The client for this project is Creative InfoTech Solution Private Limited which is situated in Parasi, Nawalparasi. It is IT company & tuition centre providing services more than 3 years which has huge number of employees and students.

Selected client Name: **Creative InfoTech Solution Pvt. Ltd.**

Address: Ramgram-1, Nawalparasi

E-mail: creativeinfotechsolution683@gmail.com

Phone: +977-9840013424

Client's Logo:



Figure 3: Client's Logo

The questionaries with its response of client about the project is illustrated in the following figure:

## Client meeting about the project

**Client Name:** Creative Infotech Solution Pvt. Ltd.

**Application:** Face Recognition Attendance System

**Agenda:** Meeting on Application Requirement

### Q&A Session

1. What is the general process of recording attendance information of your student?  
⇒ Actually, we've been using attendance manual in register.
2. What is the sole purpose of your interest in this particular project?  
⇒ Recording attendance in register seem outdated and time consuming so we decided to upgrade to this project.
3. How can it be beneficial for your company?  
⇒ Because of its less time consuming and auto system.
4. Should student need account and access to attendance?  
⇒ We consider providing no any access to student, all the control must be over the admin.
5. What is the face recognition accuracy you prefer?  
⇒ We can consider almost 75 percent matching.
6. Do you want accuracy percentage displaying in screen?  
⇒ That would be great if its displays.
7. Do you prefer digital clock display in the application?  
⇒ Yeah, sure. We would appreciate it.
8. Should we provide email confirmation feature for new admin registration?  
⇒ That might not be necessary, cause only limited number of admin might have this application.
9. Which database would you prefer to store data?  
⇒ MySQL database seem much better than other at this application.
10. Would you prefer attendance to store in MS Excel or just comma-separated values (CSV) file?  
⇒ Just CSV file would be enough for us.
11. Would you prefer to send mail to absentees?  
⇒ That would not be necessary, we would make phone call for three continuous absent case.

Figure 4: Client Meeting with feedback

### 2.1.2 Client approval letter

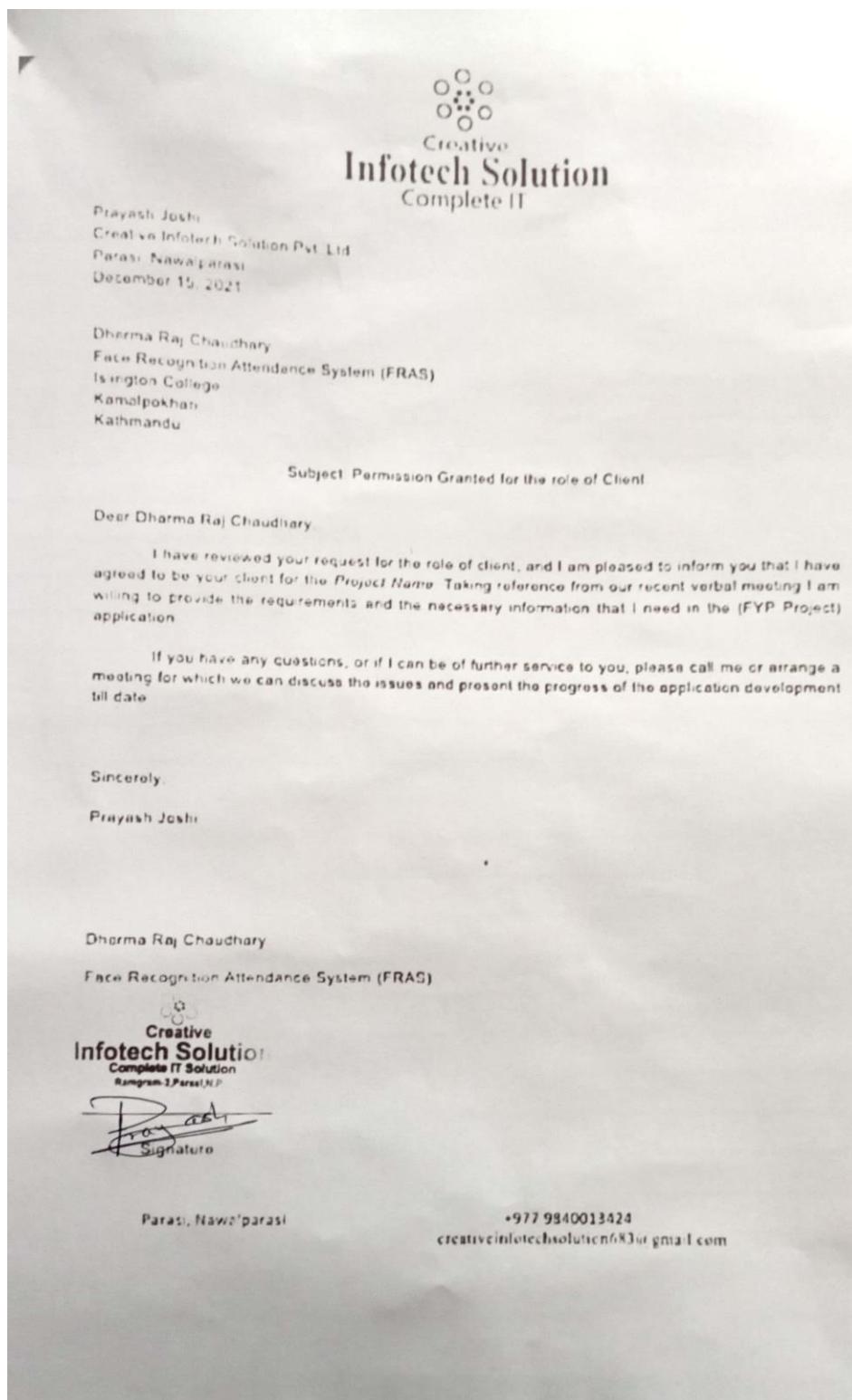


Figure 5: Client Approval Letter

### 2.1.3 Discussion about the project with client

Through the discussion and the meeting held with the client, Mr. Prayash Joshi, managing director of Creative Infotech Solution, according to him, the problems are:

#### 2.1.3.1 Problem

According to him, student information and data, as well as staff data, are stored manually, and manually storing data is a very time-consuming task. There is a waste of time and no consolidated information system as a result. He had been considering putting in place a system for his company where students and employees could easily take attendance with little effort and in a short amount of time, and where the attendance could be stored in the system itself, making it more feasible for students and employees, as well as making it easier to store data in the system.

#### 2.1.3.2 Solution

Students will be able to submit attendance using the system as part of this project. The features will be deployed in accordance with the client's meeting. This system allows you to add, remove, or update information about your students. The information or details of staff and students can be stored and added to the system, giving the teacher or lecturer convenient access. Only the teacher has complete control over the system. The attendance file (csv) can be imported and exported with some changes.

Moreover, the client's requirements are described in [1.5 client's requirement](#) section.

## 2.2 Understanding the Solution

### 2.2.1 IoT as a Platform

IoT platforms give enterprises, developers, and users a leg up on the competition by including built-in tools and features that make IoT easier and less expensive. IoT platforms are an important part of the IoT ecosystem and a rapidly developing market, with a market value of more than \$22 billion predicted by 2023. Businesses may save money on development, launch faster, and improve operations by using IoT platforms (McClelland, 2020).

IoT platforms are often used to perform ongoing administration chores and data visualization, allowing users to automate their surroundings. These platforms can be thought of as a link between the data acquired at the edge and the user-facing SaaS or mobile app. Middleware solutions, which are the "plumbing" of the IoT, are often referred to as IoT platforms. In general, an Internet of Things (IoT) or Machine-to-Machine (M2M) solution is a mashup of functionality from several providers (Link Labs, 2016).

### 2.2.2 Project Elaboration

In context of this project, basically human face or image is captured through use of camera after which computer algorithms are used in facial recognition systems to pick out specific, distinguishing features on a person's face. These features, such as eye distance or chin shape, are then transformed to a mathematical representation and compared to data on other faces in a face recognition database. A face template is data about a specific person's face that differs from a photograph in that it is meant to only include certain traits that may be used to differentiate one person from another. When it matches faces then it displays either ID and name of person on the computer display after which it marks present or absent based on recognition which is pre-set into programs algorithms. While there is just admin

panel where admin can enroll new person and train code through GUI provided. After attendance completion, admin have access to preview and modify attendance sheet stored as file.

### 2.2.3 Project Deliveries

This project targets upgrade attendance system from manual(traditional) paperwork attendance system to automated attendance through face recognition attendance. This project might go well in huge company or campus having numerous hundreds or thousands of employees or students. This project simply makes attendance system comparatively faster which reduce time consuming traditional attendance system. As data is stored in database, attendance can be previewed even after decades of years. Simply paperwork can get wear and tear after few years but it comes pros while data can be stored in cloud database too which prevent risk of wear and tear.

## 2.3 System architecture

### 2.3.1 System architecture of Face recognition attendance system

The system comprises of a camera that captures photos of the classroom and compares them to the student dataset. Once a match is made, the system notes the student's attendance and exports the data to a database. The system architecture for the Face Recognition Attendance System is depicted in the diagram below.

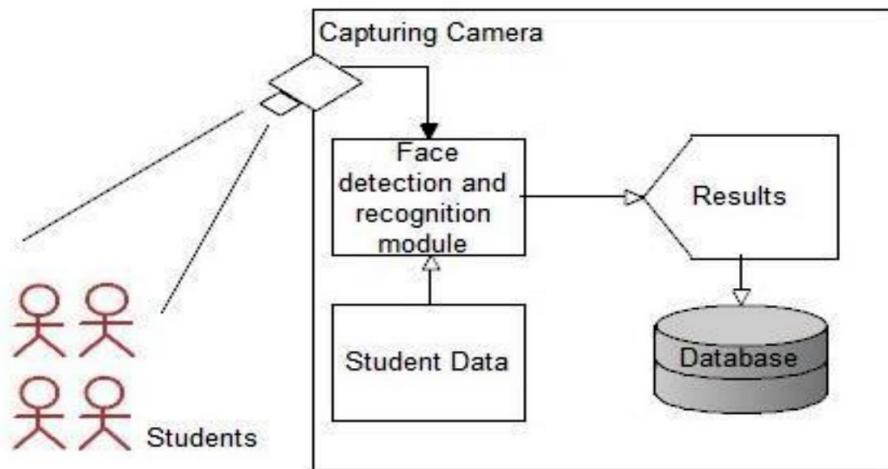


Figure 6: System Architecture

### 2.3.2 Project Overview Architecture

## 2.4 Tools and Technologies

### 2.4.1 Xampp (Mysql Server): Database

XAMPP is a set of free tools for setting up and running the Apache Web server. phpMyAdmin is an Apache package that includes the Apache Web server, MySQL, PHP, Perl, an FTP server, and the Apache Web server. For Linux, Solaris, and Windows, XAMPP is available. "The aim of XAMPP is to build an easy-to-install distribution for developers to come into the world of Apache," according to the Apache Friends Web site (McCarthy, 2005)." In this project, Xampp is used to create a MYSQL Server database that stores all of the registration and student data under the name "face recognizer." MySQL is the most widely used relational database management system (RDBMS), and it has a client-server architecture.



Figure 7: XAMPP Logo

#### 2.4.2 IDE: PyCharm

PyCharm is a very popular python IDE which is a cross-platform IDE that was developed by JetBrains to use it for python development. Many people these days believe that python is the best language where a user can build software applications by writing clean and readable code. Python is the favorite language for many, especially people working in Data Science and Machine Learning<sup>1</sup> (Dwivedi, 2020).



Figure 8: PyCharm Logo

#### 2.4.3 Python Programming Language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in

source or binary form without charge for all major platforms, and can be freely distributed (Python Software Foundation , 2022).

- **Installation-**

```
$ sudo apt-get install python<version>
$ brew install python<version>
$ pip install python<version>
```

#### 2.4.4 GUI Framework: Tkinter

Tkinter is Python's standard GUI library. When Python is used in conjunction with Tkinter, creating graphical user interfaces is quick and simple. The Tk GUI toolkit has a sophisticated object-oriented interface called Tkinter. Python has a lot of GUI frameworks, but Tkinter is the only framework that's built into the Python standard library. Tkinter has several strengths. It's cross-platform, so the same code works on Windows, macOS, and Linux. Visual elements are rendered using native operating system elements, so applications built with Tkinter look like they belong on the platform where they're run (Amos, 2022).

The window is the most basic component of a Tkinter GUI. The containers in which all other GUI elements exist are known as Windows. Widgets are the other GUI elements such as text boxes, labels, and buttons. Widgets are small pieces of software that are installed on the inside of windows.

- **Installation**

```
Python 3.0 - $ sudo apt-get install python3-tk
Python 2.7 - $ sudo apt-get install python-tk
```

### 2.4.5 Python Libraries

A Python library is a reusable code snippet that you can use in your programs and projects. Python's Standard Library is a collection of the language's exact syntax, tokens, and semantics. It's included in the standard Python distribution. There are bunch of python libraries used and implemented in this project which are brief elaborated in [appendix B](#).

### 2.4.6 Classifier and Algorithms

#### 2.4.6.1 Face Recognition Using HAAR-Cascade

This approach is based on the HAAR-Cascade technique to analyze pixels in the image into squares by function. It is a machine learning approach where a cascade function is trained from multiple positive and negative images from the training data. HAAR feature extraction filters which are as shown below, are used to detect various informative components of the face which help define the features of the particular individual (International Journal of Innovative Research in Science, Engineering and Technology, 2019).

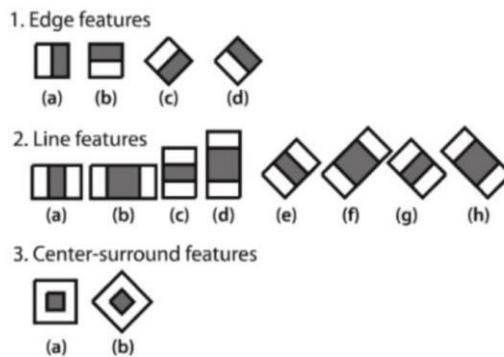


Figure 9: Face Feature

Each feature is a single value calculated by subtracting the sum of pixels beneath the white rectangle from the total of pixels beneath the black rectangle.

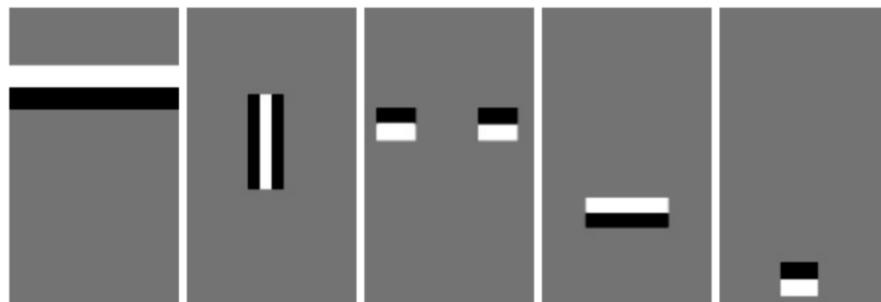


Figure 10: Face Feature ii

Each characteristic is applied to all of the training photos. It determines the appropriate threshold for each feature to identify the faces as positive or negative. Face and non-face photos are accurately classified using features with a low error rate (International Journal of Innovative Research in Science, Engineering and Technology, 2019).



Figure 11: HAAR filter Overlay on Face

The non-face region of an image makes up the majority of the image. As a result, having a simple method to check if a window is not a face region is a better approach. If it isn't, toss it out in one go and don't bother processing it again. Instead, concentrate on areas where a face might appear. We can spend more time checking possible face regions this way (International Journal of Innovative Research in Science, Engineering and Technology, 2019).

Instead of applying all thousands of characteristics to a window, Cascade of Classifiers is employed, which divides features into distinct stages of classifiers, which are then applied one at a time. If a window fails the first step, it and its remaining features are discarded. The second level of characteristics is applied if it passes. This procedure is still in progress. A face region is defined as a window that has passed all of the stages (International Journal of Innovative Research in Science, Engineering and Technology, 2019).

#### 2.4.6.2 Local Binary Pattern Histogram Face Recognizer (LBPH) Classifier

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. The LBPH Face Recognizer Process – Take a  $3\times 3$  window and move it across one image. At each move (each local part of the picture), compare the pixel at the center, with its surrounding pixels. Denote the neighbors with intensity value less than or equal to the center pixel by 1 and the rest by 0. After you read these 0/1 values under the  $3\times 3$  window in a serial order, you will have a binary pattern like 10001101 that is local to a particular area of the picture. When you finish doing this on the whole image, you will have a list of local binary patterns (International Journal of Innovative Research in Science, 2019).

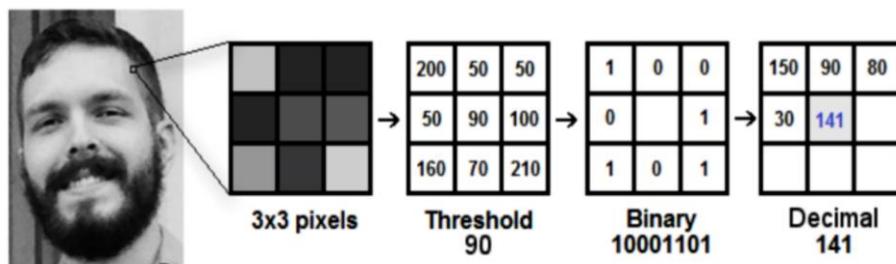


Figure 12: Sample 3x3 Grid of a Recognizer

Binary conversion of LBP. Local Binary Patterns Applied to Face Detection and Recognition, López & Ruiz

After obtaining a list of local binary patterns, you convert each one into a decimal number (as illustrated in the above figure) and then create a histogram of all of those decimal values. This is an example of a histogram:

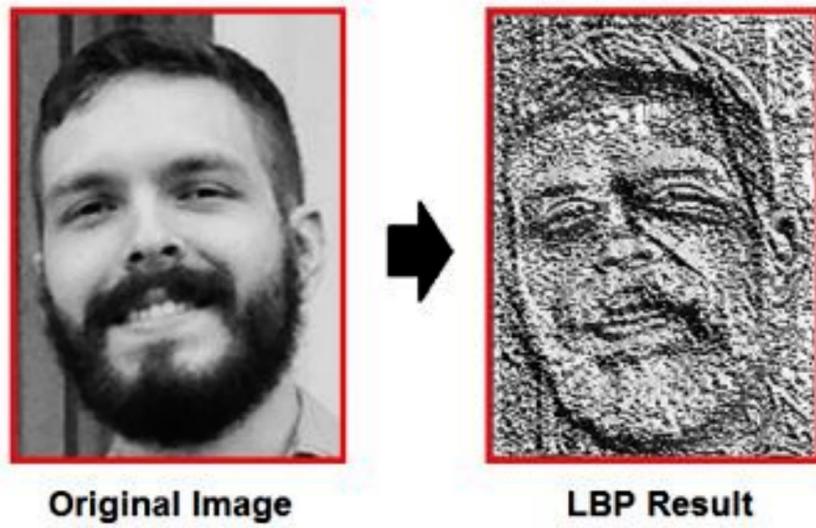


Figure 13: Output of LBPH from original image

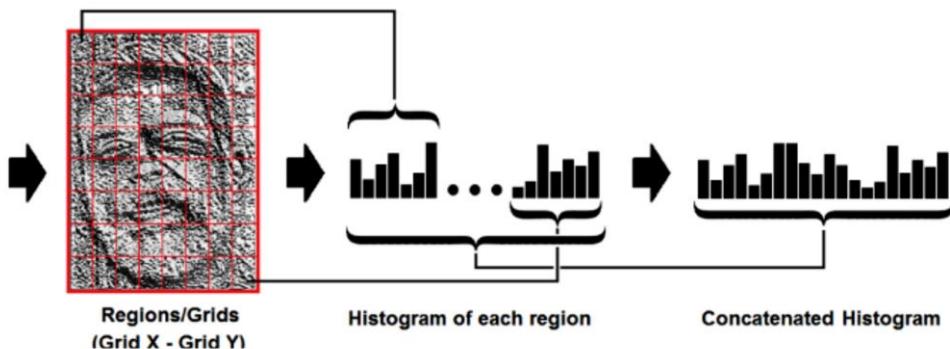


Figure 14: Histogram Representation of LBPH Result

In the end, each face in the training data set will have its own histogram. That is, if the training data set contains 100 photos, LBPH will extract 100 histograms after training and store them for later recognition. In addition, the system keeps track of which histogram

corresponds to which individual. Light affects both Eigenfaces and Fisherfaces, and perfect light conditions are not always accessible in real life. The LBPH face recognizer is a step forward in overcoming this flaw. A group of faces and their local binary pattern images are shown below. As you can see, changes in light have little effect on the LBP faces (International Journal of Innovative Research in Science, 2019).

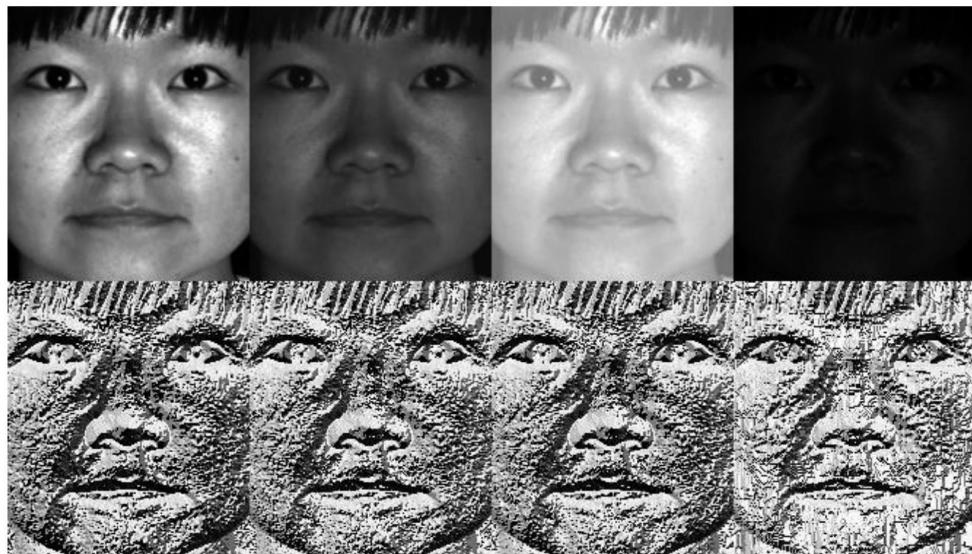


Figure 15: Effect on Lighting on LBPH Faces

## 2.5 Similar Projects

### 2.5.1 Project 1: Smart Student Attendance System Using QR Code

In today's rapidly changing technological environment, QR codes have a wide range of applications. QR codes are used to store large amounts of data in a little amount of area. As a result, QR codes are chosen to incorporate into system and offered a smart attendance system based on QR codes. Data-hiding methods and the embedded QR Code are used to provide secure authentication. In our initiative, students scan QR codes with their smartphones, which are then shown by the teacher. When students scan this QR code, their attendance is

immediately recorded based on their user id. It also explains how the system verifies student identity in order to prevent fraudulent registrations (Arpankumar Patel, 2019).

### Pros of this system:

- Attendance data that are timelier and more accurate
- Front desk is less busy
- Parents and students will have more face-to-face time.

### Cons of this system:

- Removes absences and restores functionality
- Changes the focus from "monitoring absences/make-ups" to "tracking attendance."

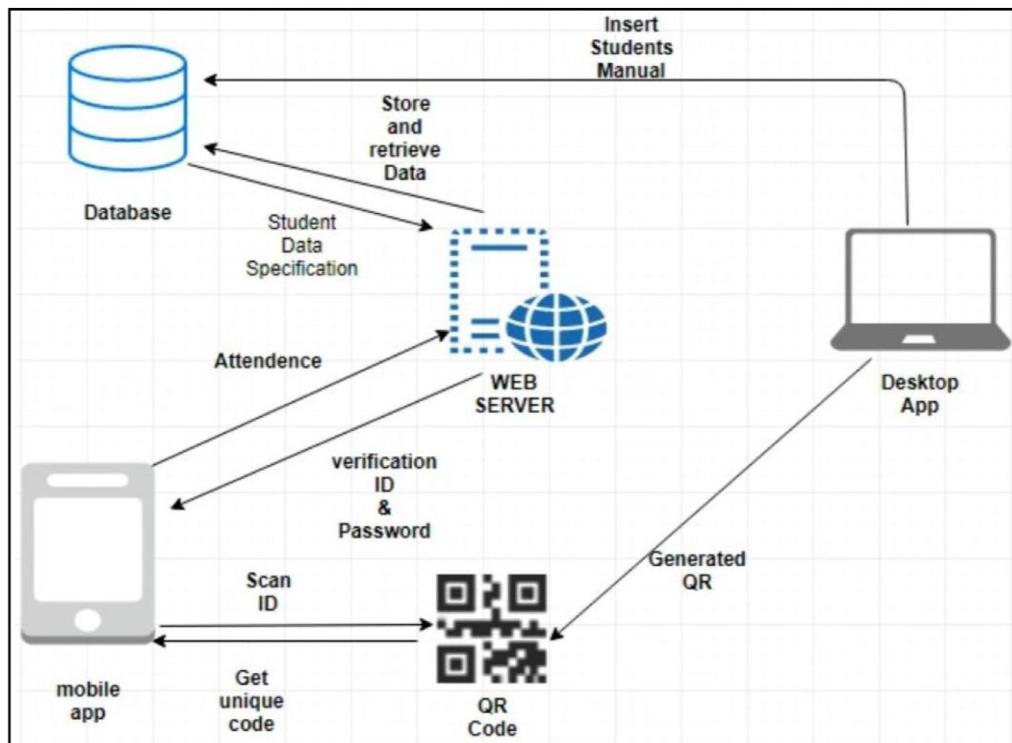


Figure 16: Smart Student Attendance System Using QR Code (Elbehiery, 2019)

### 2.5.2 Project 2: Smart Attendance Monitoring System Using IoT And RFID

Maintaining the attendance of the students in an institution is a heavy task. Always there is a difficulty in handling attendance manually. This project aims at designing a smart attendance system that automatically monitors and manages attendance of the students in an institution efficiently. The whole system is developed with an Arduino uno microcontroller and RFID readers. Unique RFID tags can be deployed in students' id card. Also, Wi-Fi communication modules are used to make convenient communication depend on the availability of the network. Database of students must be created. A GSM Module is used to send messages to parent's mobile about the students attendance status. A GPS module is used to detect the live location of the student. This system will reduce a lot of manual work of teachers and administrators of any institution. The proposed work comprises of two most popular trend in technology research; IoT and RFID (G.T.Bharathy, 2021).

#### **Pros of this system:**

- The System allows for more precise identification.
- Quick and efficient: Identifies prospects in a matter of seconds.
- The System is less time-consuming, cost-effective, simple to use, difficult to tamper with, and modest.

#### **Cons of this system:**

- The system is costly since it incorporates a significant amount of technology.
- Purchasing tags for all kids is expensive if there is a large group of students.
- It is time consuming and costly to replace the microchip, radio transceiver, antenna, and battery in the system.
- The system is vulnerable to manipulation because it is not as secure as biometrics.

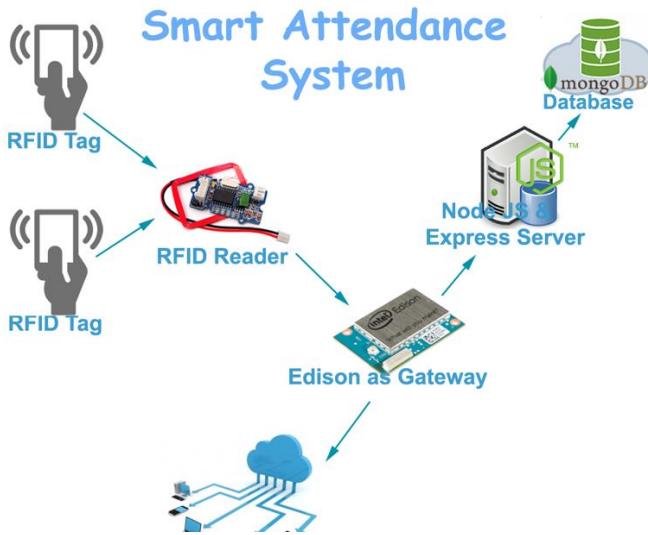


Figure 17: Smart Attendance Monitoring System Using IoT And RFID (taifur, 2018)

### 2.5.3 Project 3: Smart Attendance System Using Finger Print Biometric

In this system the attendance collection is done with the help of biometric input which is done as finger print scanning. In this system the user input is provided with the help of a finger print. Once this input enters the system the system compares the data with the data stored in database. If the data is found valid, then the system will record the attendance information. If the details are found invalid then the student attendance information is not recorded and system will exit. This system will ensure that the data entering the system is authentic since there will not be any issue of proxy attendance. This is one of the most commonly implemented system for attendance collection because of its simple but efficient nature (Muralidharan P, 2016).

#### Pros of this system:

- It is one of the oldest biometric systems.
- It is instant attendance system.

### Cons of this system:

- Using a fingerprint scanner can result in erroneous rejections.
- Computer hardware and software applications can be quite costly.
- Using a fingerprint scanner may result in incorrect approval.
- When using a fingerprint scanner, it is not taken into account when a person's physical appearance changes.

## 2.6 Comparisons

### 2.6.1 Comparison Table

S. N	Features	Project 1: QR	Project 2: RFID	Project 3: Fingerprint	This project
01	Database	✓	✓	✓	✓
02	Microcontroller used	Arduino Uno	Arduino Uno	Raspberry pi	Either or none
03	Time efficient	Medium	Medium	Medium	High
04	Cost effective	Medium	High	Medium	Low
05	Accuracy	High	High	High	High
06	Easy on use	Medium	Medium	Medium	High
07	Multi-attendance at a time	Low	Low	Low	Low
08	Tools required	QR Scanner devices	RFID Card and reader	Scanner	Camera

Table 1: Similar project comparison table

### 2.6.2 Conclusion From the Similar Projects

The analysis demonstrates that no single solution is the best fit for every company's requirement. Every system has their own advantages and disadvantages. As a result, any company can choose which system to investigate based on its requirements. A mix of one or two of the technologies employed in these systems will be required in some instances, or in most cases, to ensure that the end result is a well-performing, well-secured, and easily manageable system.

But taking all the features provided by similar projects into consideration, we can clearly see that the purposed system provides all the basic features at a very minimal cost. It does decent work with less complexity. Each other project requires other devices and are more time consuming than that proposed project. As its main motive is to reduce time consuming, this project seems best of all. As well as it is easiest to use, no student or employee's involvement requires in it as face detection is from far behind. This project is probably only which is taking attendance of multiple people at the same time which is most time efficient.

## Chapter 3: Development

### 3. Development

#### 3.1 Considered Methodology

The numerous software development approaches available provide developers with a plethora of options. The majority of techniques fall into one of three categories: waterfall, iterative, or continuous.

Some of the considered methodologies are presented in [appendix A](#).

#### 3.2 Selected Methodology: Evolutionary Prototyping

Before we get into what an evolving prototype is, it's important to understand what a prototype is and why it's needed. A prototype is a method of creating a product or system that closely resembles or duplicates the real thing. A product prototype is used to gather client input in order to improve the product's design and functioning before it goes live. In the software industry, there are numerous prototyping models utilized, including quick throwaway prototyping, incremental, extreme, and evolutionary prototypes (Martinez, 2020).

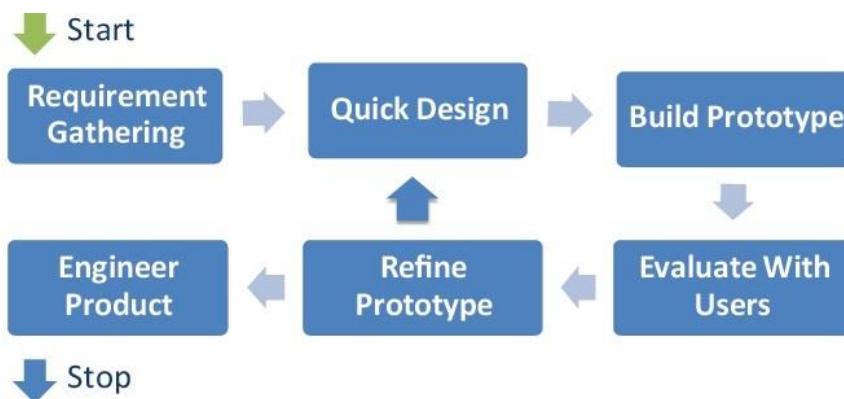


Figure 18: Evolutionary Prototype

Because changes are a common and necessary element of the system, Evolutionary Prototype technique was used for the system. To make the project more practical, efficient, and

market-ready, new features could be added or an existing feature could be deleted. In the IoT system, new ideas are common, and with new ideas come new adjustments, allowing the system to be flexible. Design, testing, and improvement are what allow for innovation and modifications that will eventually please the customer. Because the client's requirements are unclear from a project standpoint and may vary over time, the prototype can be altered several times. In the context of this system, the evolutionary prototype process also encourages criticism from both internal and external oversight.

- The design of this model is adaptable.
- Error is easy to spot.
- Lacking functionality can be quickly located.
- There is room for improvement, implying that new requirement can be accommodated quickly.
- It can be utilized by the developer in the future for more complex applications.
- It ensures that customers are more satisfied and comfortable.
- This type of approach of developing the software is used for non-IT-literate people.

They usually are not good at specifying their requirements, nor can talk properly about what they expect from the software.

Some disadvantages of Evolutionary Prototype Methodology are enlisted below:

- The number of iterations can be calculated with confidence.
- It's possible that the problem analysis is incomplete or insufficient.
- The system's complexity may increase as a result.

- This model is costly.
- It is slow process.

### 3.3 Comparison Between Any Two Methodologies

Features	Evolutionary Prototype Model (Selected Methodology)	Iterative Waterfall Model (Considered Methodology)
Requirements	Requirements can change frequently while developing the system.	Requirements are gathered at the beginning and can be changed only at the beginning of the iteration.
User Involvement	There is high user involvement.	There is medium user involvement.
System Complexity	The system can get complex according to the requirement of the user.	A simple system will be developed.
Successful Completion	There is an average chance for successful development of the system.	There is a high chance of successful development of the system.
Overlapping of phases	There will be overlapping of phases in the prototyping model.	There will not be any overlapping of phases.
Cost	High cost	Low cost
Cost Control	Cost control is not possible	Cost control is possible

Table 2: Comparison between any two methodologies

### 3.4 Similarity Between Evolutionary Prototype Model and Iterative Waterfall Model

Feature	Evolutionary Prototype Model (Selected Methodology)	Iterative Waterfall Model (Considered Methodology)
Availability of reusable components	Yes	Yes
Risk analysis	No Risk analysis	No risk analysis
Changes incorporated	Easy	Easy
Cost Control	No	No

Table 3: Similarity between evolutionary and iterative waterfall model

### 3.5 Phases of Methodology

This evolutionary prototyping methodology is categorized into six stages which are elaborated below: -

#### 3.5.1 Phases of Prototype

##### 3.5.1.1 Requirement Gathering and Analysis

The client was scheduled for a meeting to obtain requirements, and the requirements were understood. All of the requirements were meticulously documented. This step identifies the information that will be included in the application.

The requirement gathered from the client is presented in the [section 1.5](#).

##### 3.5.1.2 Quick Design

The project's wireframe and graphical design are created during the design phase. In addition, the use case and entity relationship diagram were created. Balsamiq is used to create

wireframes and drawings. Gantt project is used to create Gantt charts, while io is used to create use cases, entity relationship diagrams, and work breakdown structures.

The diagrams which had been developed in this phased are illustrated in [section 3.8.1.3.](#)

### **3.5.1.3 Build prototype**

The draft design was developed and then followed with development process.

### **3.5.1.4 Initial user evaluation**

A meeting with the client has been scheduled to discuss the first phase prototype. The initial prototype has been deployed and is available for use by the customer. The client looked through the prototype and gave input on the requirements that were gathered and developed. And, as previously said, this process should be repeated until the final product is launched.

### **3.5.1.5 Refining prototype**

It is necessary to make the necessary modifications. A final prototype is created with the client's approval.

### **3.5.1.6 Implement and maintain**

The final product is conceived and developed when the client approves the prototype. Multiple tests are carried out with fewer failures. The client approves all of the application's features and the general appearance.

### 3.6 System Requirements Analysis

#### 3.6.1 Technical Requirement

##### 3.6.1.1 Hardware Requirement

- A Standalone Computer (4GB RAM or higher)
- High-quality camera
- Additional memory for storing all of the photographs and the database.

##### 3.6.1.2 Software Requirement

- MYSQL Server (XAMPP)
- PyCharm Community (Version: 2021.3.3 or above)
- Linux Based OS or Windows 8 or higher
- Python 3.9 or above
- Latest version of all libraries

#### 3.6.2 Functional Requirement

The term "functional requirement" refers to the activities and services that must be provided by the system.

- An administrator must be able to manage student/employee records
- The system must be used by an authorized user.
- The system must be linked to a camera, and face recognition must be accurate.
- Before using the system, the administrator or the person who will be given access to it must log in.
- The data must be correctly entered and managed.
- Input validation must be maintained.

### 3.6.3 Non-Functional Requirement

Non-functional requirements are characteristic of attribute of the system that can judge its operation. The following points clarify them all:

- Accuracy and precision: To avoid issues, the system should conduct its process with accuracy and precision.
- Flexibility: The system should be simple to change, and any errors should be corrected.
- Security: The system should be safe and secure, protecting the privacy of students.
- Usability: The system should be simple to operate and comprehend.
- Maintainability: Any problem that arises unexpectedly should be handled by the maintenance group.

**Non-Functional requirements** are as follow:

- The GUI of the system should be user-friendly.
- The data that will be displayed to the user should be checked to ensure that it is correct and current. The system must be adaptable to change.
- Efficiency and effectiveness of the system should be made sure.
- The performance of the system should be made sure.

### 3.6.4 Admin/Teacher Requirement

- When registering a student for the face recognition process, the administrator must log into the system.
- Only the system administrator has the authority to make changes to the system.
- Only admin is allowed to manage any changes in the stored data sets.

### 3.7 Design

Use case diagrams, entity relationship diagrams, UML sequence diagrams, and flow charts of the overall project are needed to depict the general process and flow of the system.

Below are diagrams such as communication diagrams, wireframes, use case diagrams, and sequence diagrams that aggregate all of the system's features and capabilities.

#### 3.7.1 Use Case Diagram

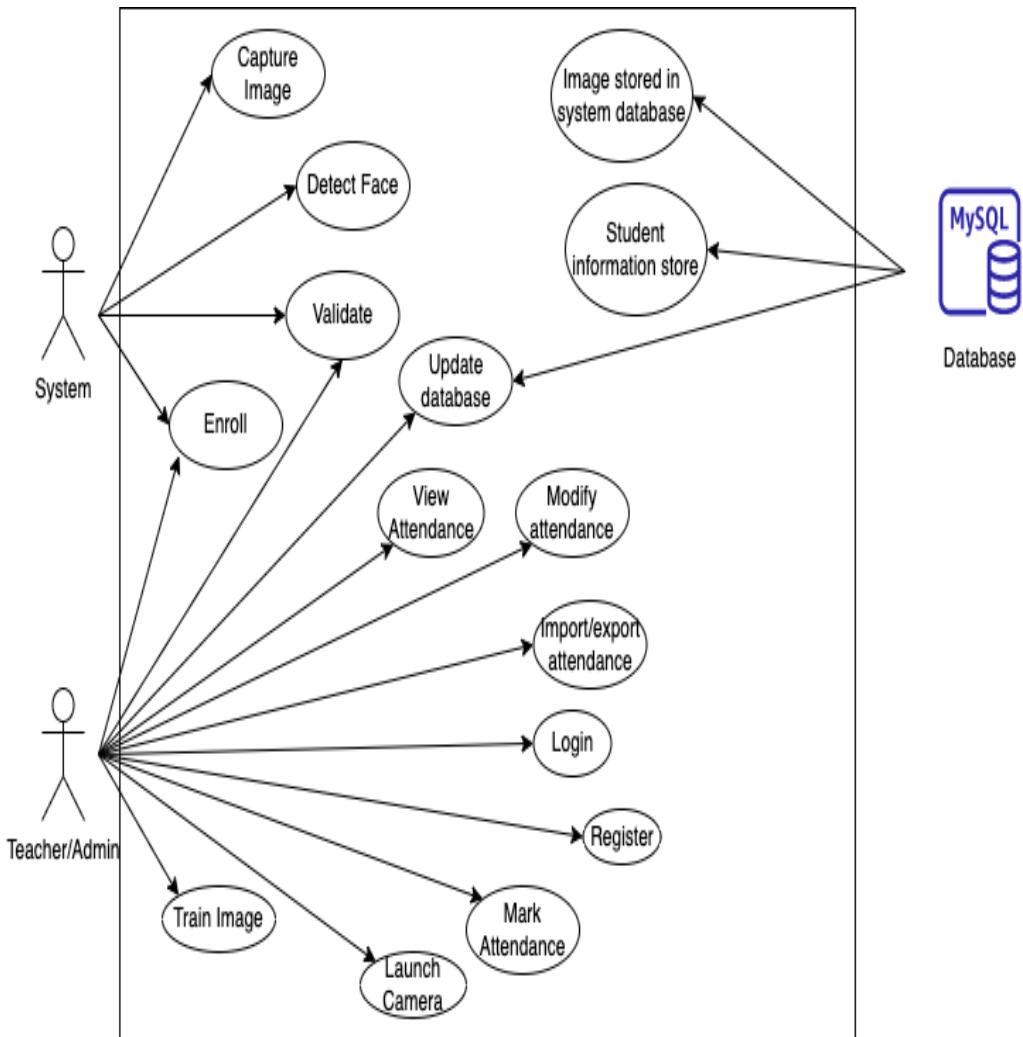


Figure 19: USE Case Diagram

### 3.7.2 UML Sequence Diagram

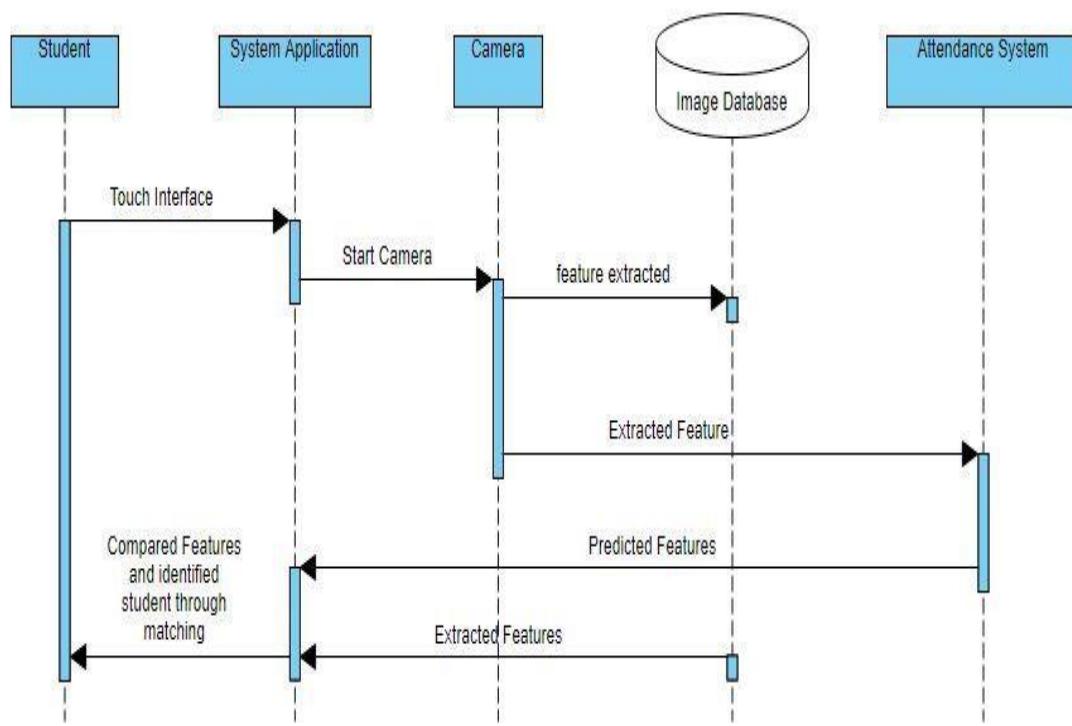


Figure 20: UML Sequence Diagram

### 3.7.3 Flow Chart

#### 3.7.3.1 Admin Flowchart

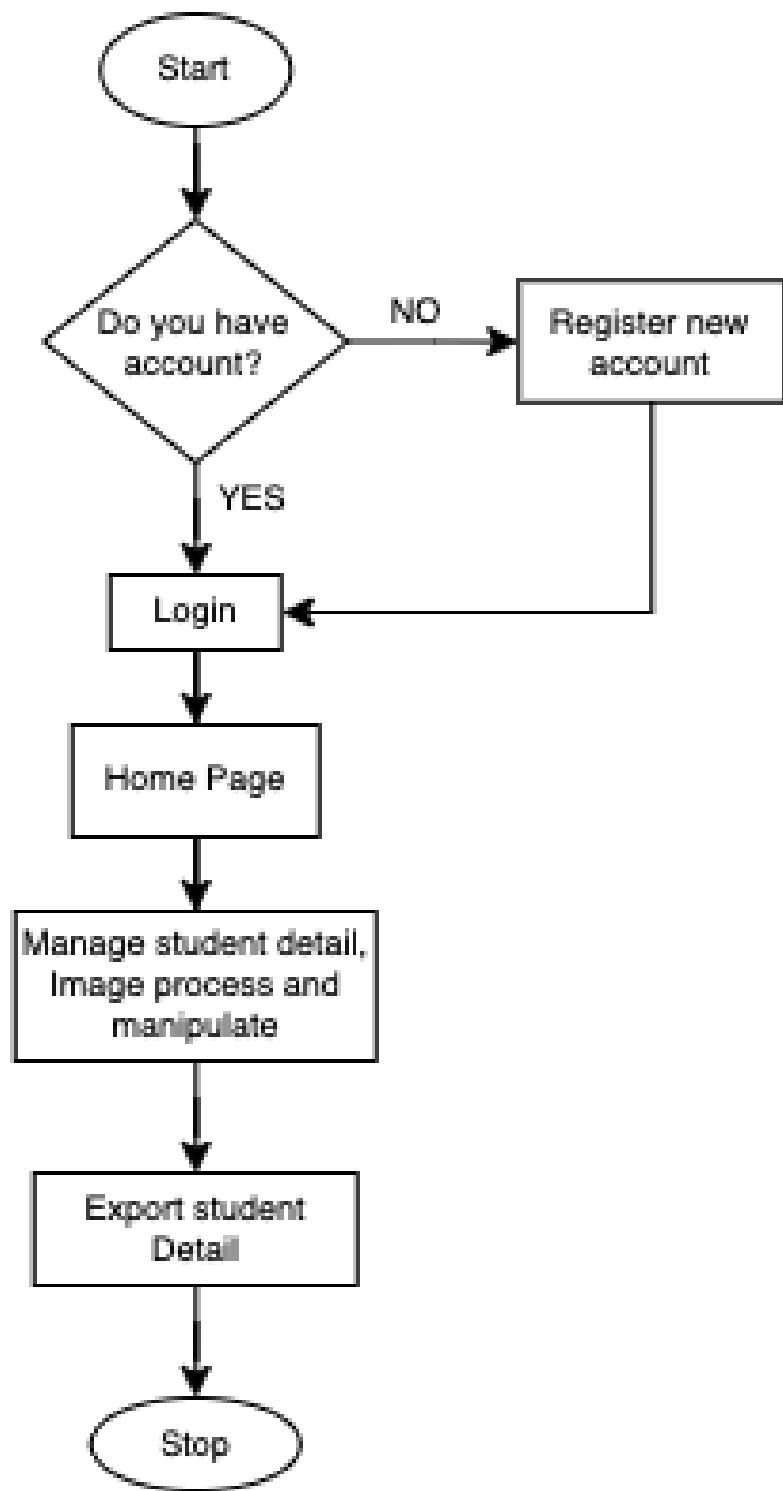


Figure 21: Admin Flowchart

### 3.7.3.2 Attendance Flowchart

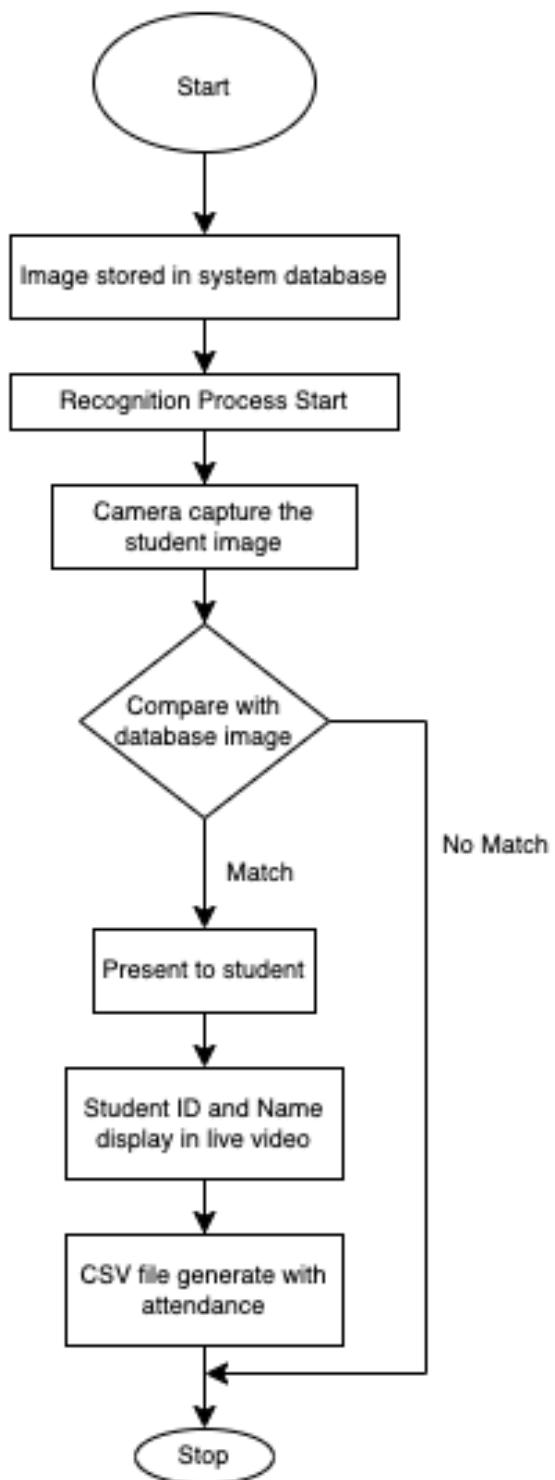


Figure 22: Attendance Flowchart

### 3.7.4 Entity Relational Diagram

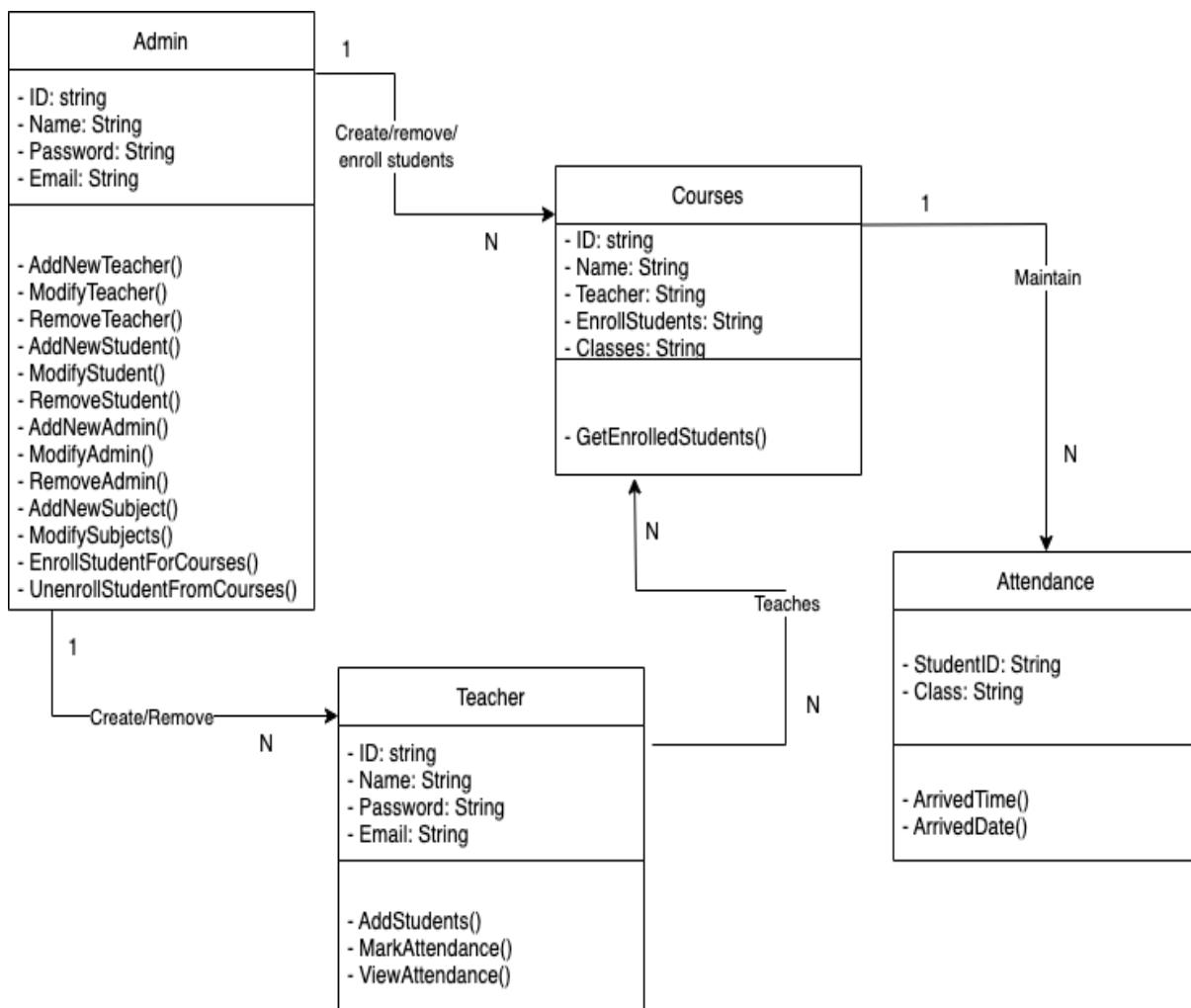


Figure 23: Entity Relational Diagram

### 3.8 Implementation of methodology

#### 3.8.1 Iteration:

##### 3.8.1.1 Requirement Gathering

The client's initial needs are collected and gathered during this phase of the approach.

As a result, I obtained the needs from my client. The requirements for the Face Recognition Attendance System were acquired through research on many websites as well as by using them to understand the application's general work flow. A survey was done to have a better understanding of the application's overall workflow. A poll was done to learn more about how individuals feel about the system.

##### Requirement Gathered:

- Admin/Client: User login, add student, manage the student, take attendance, view attendance, manage attendance, etc.

##### 3.8.1.2 Quick Design

Following the completion of the requirement collection, we defined user roles for our final system in accordance with the client's wishes. A quick design for the admin job was produced in the initial iteration. A simple design of the system for the admin's role is shown below.

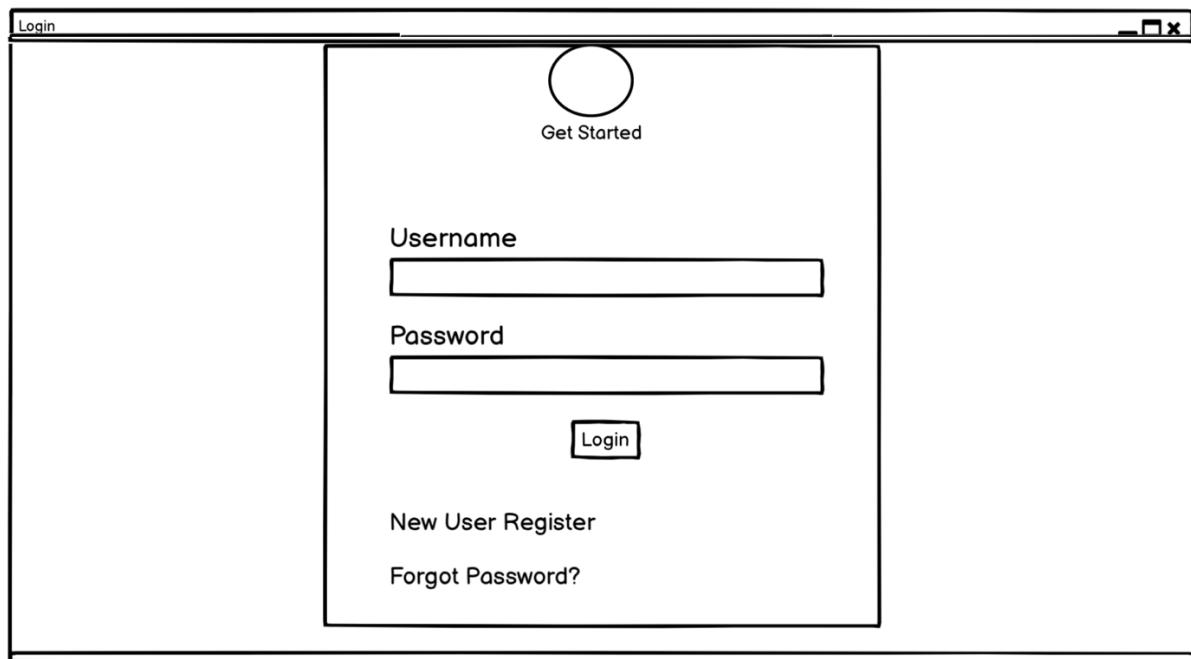
### 3.8.1.3 Build a prototype

We were able to construct a prototype for the admin's role based on the study of quick design for the admin's role. The following are fragments from the prototype/wireframes for the admin role:

The wireframe shows a registration form titled "Register Now". It includes fields for First Name, Last Name, Contact No., Email, Password, Confirm Password, Security Question, and Security Answer. There is also a checkbox for accepting terms and conditions and a "Register Now" button. On the left side, there is a large empty area with a "Create New Image" button.

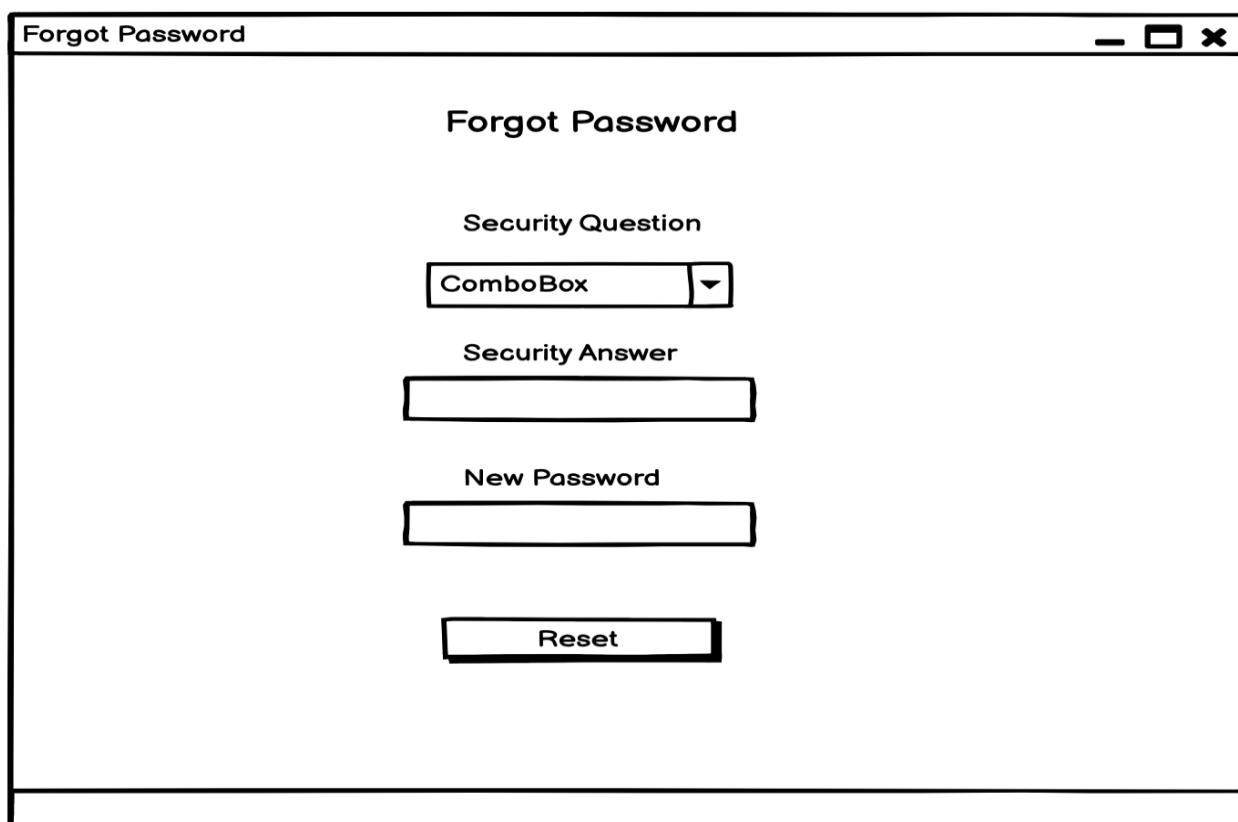
Register Now	
First Name	Last Name
<input type="text"/>	<input type="text"/>
Contact No.	Email
<input type="text"/>	<input type="text"/>
Password	Confirm Password
<input type="text"/>	<input type="text"/>
Security Question	Security Answer
Select <input type="button" value="▼"/>	<input type="text"/>
<input type="checkbox"/> I agree all the term & conditions	
<input type="button" value="Register Now"/>	

Figure 24: Wireframe - Login Window



The wireframe for the Login window shows a title bar with the word "Login". The main content area contains a circular "Get Started" button at the top. Below it are two input fields labeled "Username" and "Password". A "Login" button is positioned between them. At the bottom of the window are two links: "New User Register" and "Forgot Password?".

Figure 25: Wireframe – Login Window



The wireframe for the Forgot Password window shows a title bar with the words "Forgot Password". The main content area features a heading "Forgot Password" followed by a "Security Question" section containing a "ComboBox" with a dropdown arrow. Below it is a "Security Answer" input field. Further down is a "New Password" input field. At the bottom of the window is a "Reset" button.

Figure 26: Wireframe – Forgot Password

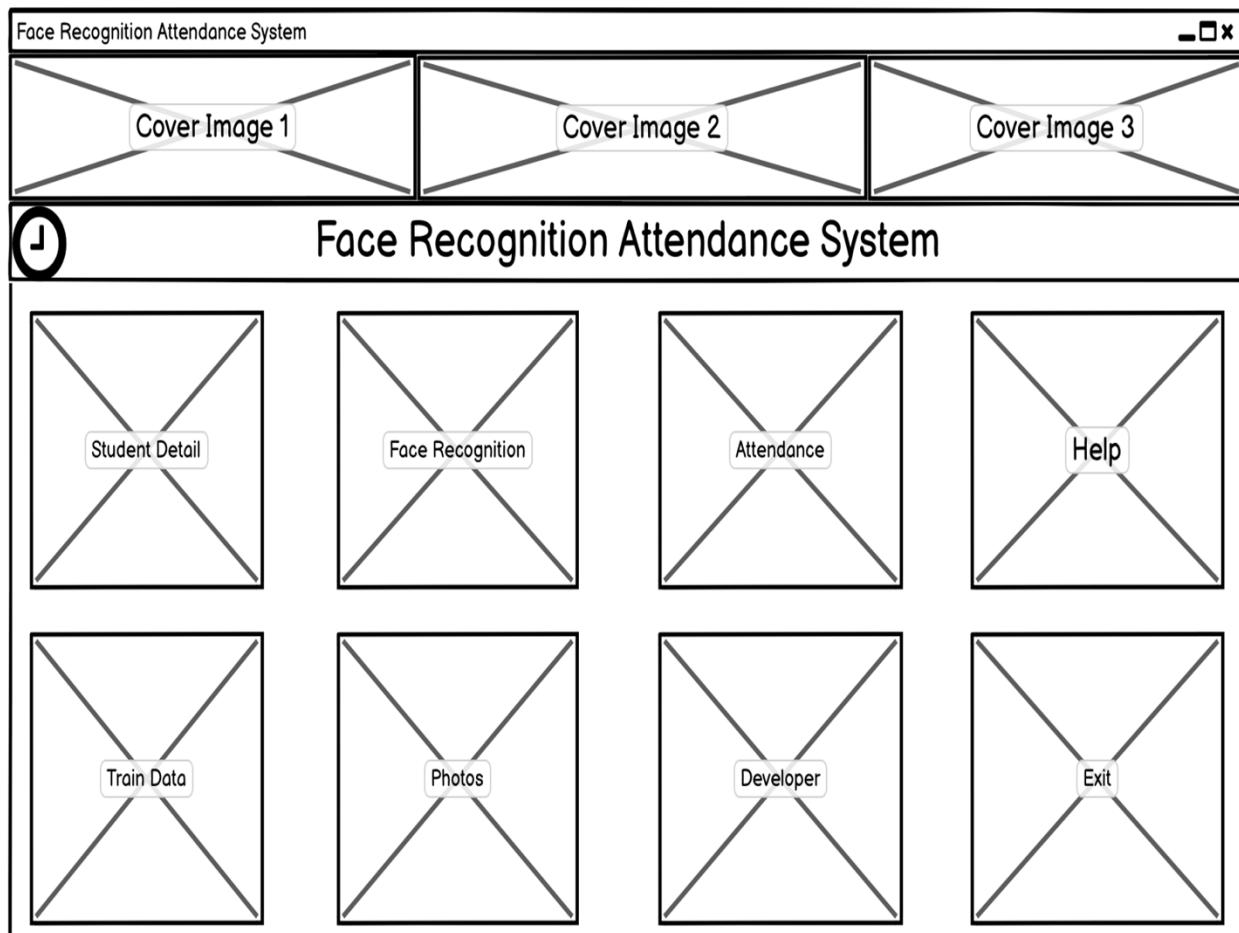


Figure 27: Wireframe – Home Window

The wireframe illustrates the layout of a Student Management System window. At the top, there are three placeholder boxes labeled "Cover Image 1", "Cover Image 2", and "Cover Image 3". Below these is a title bar with the text "Student Management System".

The window is divided into two main sections:

- Current Course Information:** This section contains four dropdown menus: "Faculty" (with "ComboBox" label), "Courses" (with "ComboBox" label), "Year" (with "ComboBox" label), and "Semester" (with "ComboBox" label).
- View Student Details & Search System:** This section features a table header with columns: "StudentID", "Student Name", "Roll No", "Gender", "Faculty", "Course", "Semester", "Year", and "Class Div". Below the header, there is a large, empty rectangular area with scroll bars on the right and bottom, intended for displaying student details.

**Student Class Information:** This section contains several input fields and buttons. It includes pairs of labels and input fields such as "Student ID" and "Student Name", "Class Division" and "Roll No", "Gender" and "DOB", "Email" and "Phone Number", and "Address".

At the bottom of this section are two radio buttons: "Take Photo Sample" and "No Photo Sample". Below these are four buttons: "Save", "Update", "Delete", and "Reset". Additionally, there are two more buttons: "Add Photo Sample" and "Update Photo sample".

Figure 28: Wireframe - Student Window

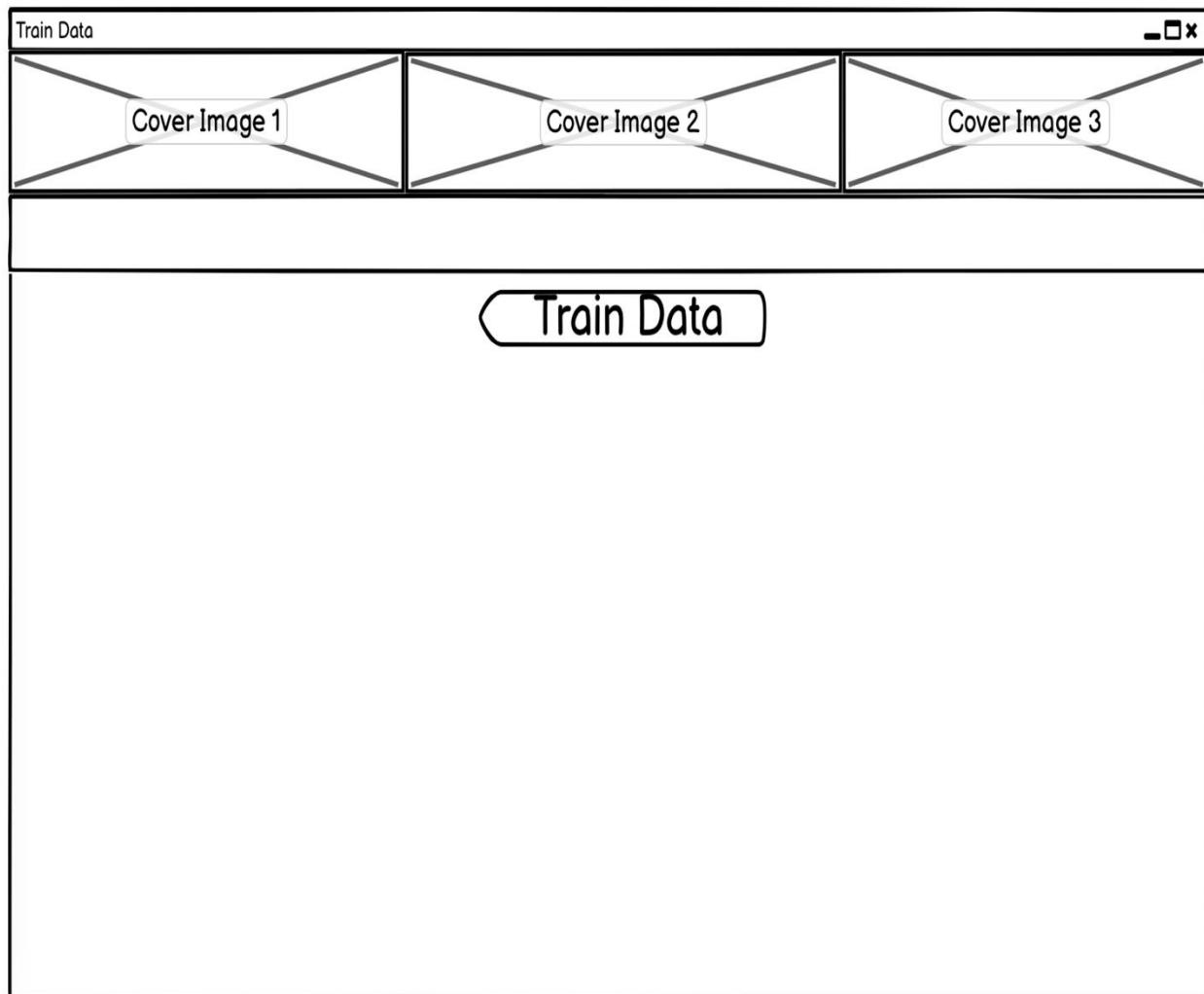


Figure 29: Wireframe - Train window

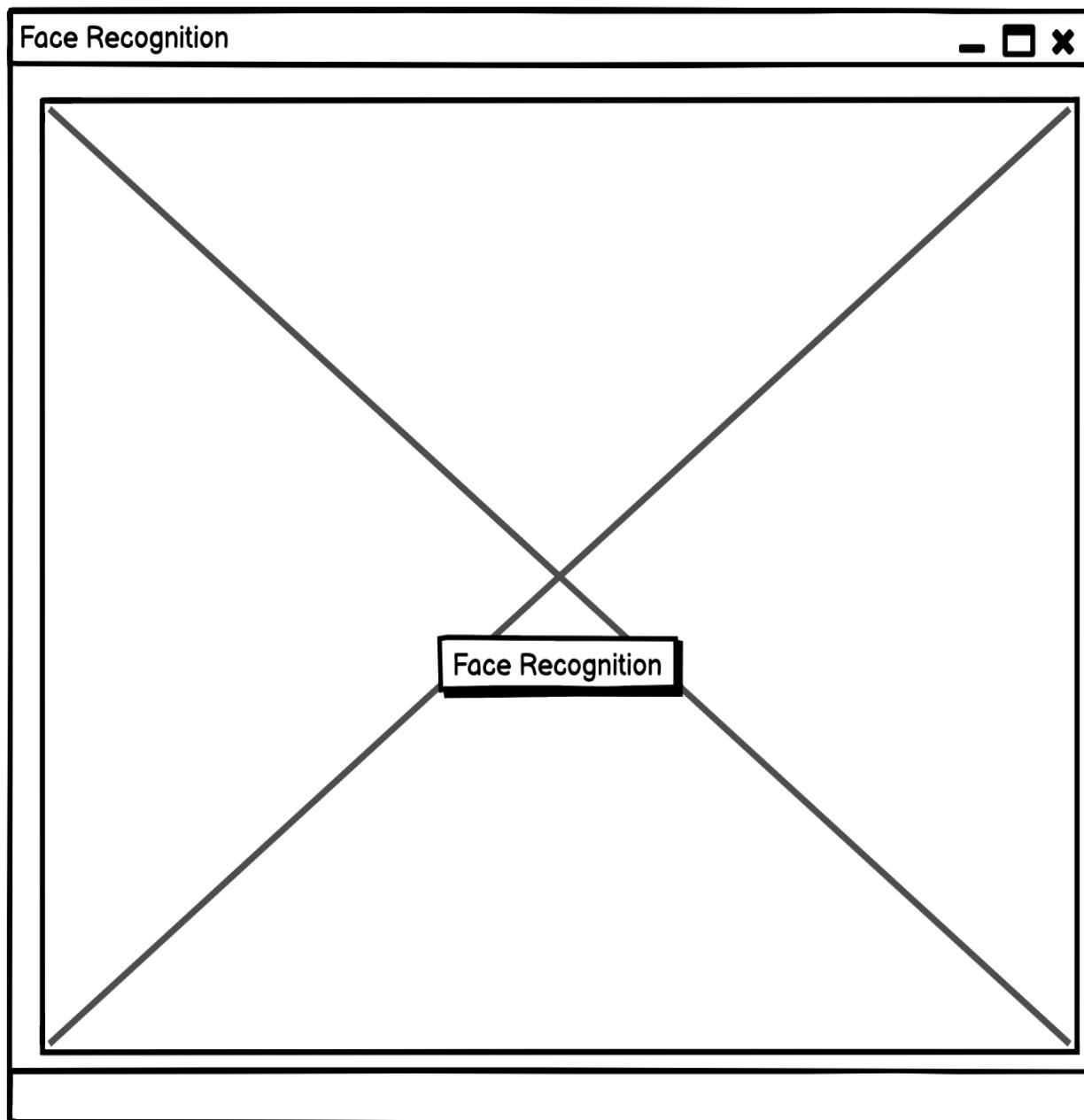


Figure 30: Wireframe - Face Recognition

The wireframe illustrates the 'Manage Attendance' window for the Face Recognition Attendance System. At the top, there are three decorative 'Cover Image' placeholders labeled 'Cover Image 1', 'Cover Image 2', and 'Cover Image 3'. Below this is the main title 'Face Recognition Attendance System'. The interface is divided into two main sections: 'Student Information' on the left and 'Student Detail' on the right.

**Student Information:**

- Student ID: [Text Input]
- Student Name: [Text Input]
- Roll No.: [Text Input]
- Attendance Status: [Text Input]
- Faculty: [Text Input]
- Time: [Text Input]
- Date: [Text Input]

**Student Detail:**

Student ID	Roll No	Student Name	Faculty	Time	Date	Attendance Status
double-click to edit						

At the bottom of the window, there are several buttons: 'Update', 'Reset', 'Import csv', and 'Export csv'.

Figure 31: Wireframe - Attendance Window

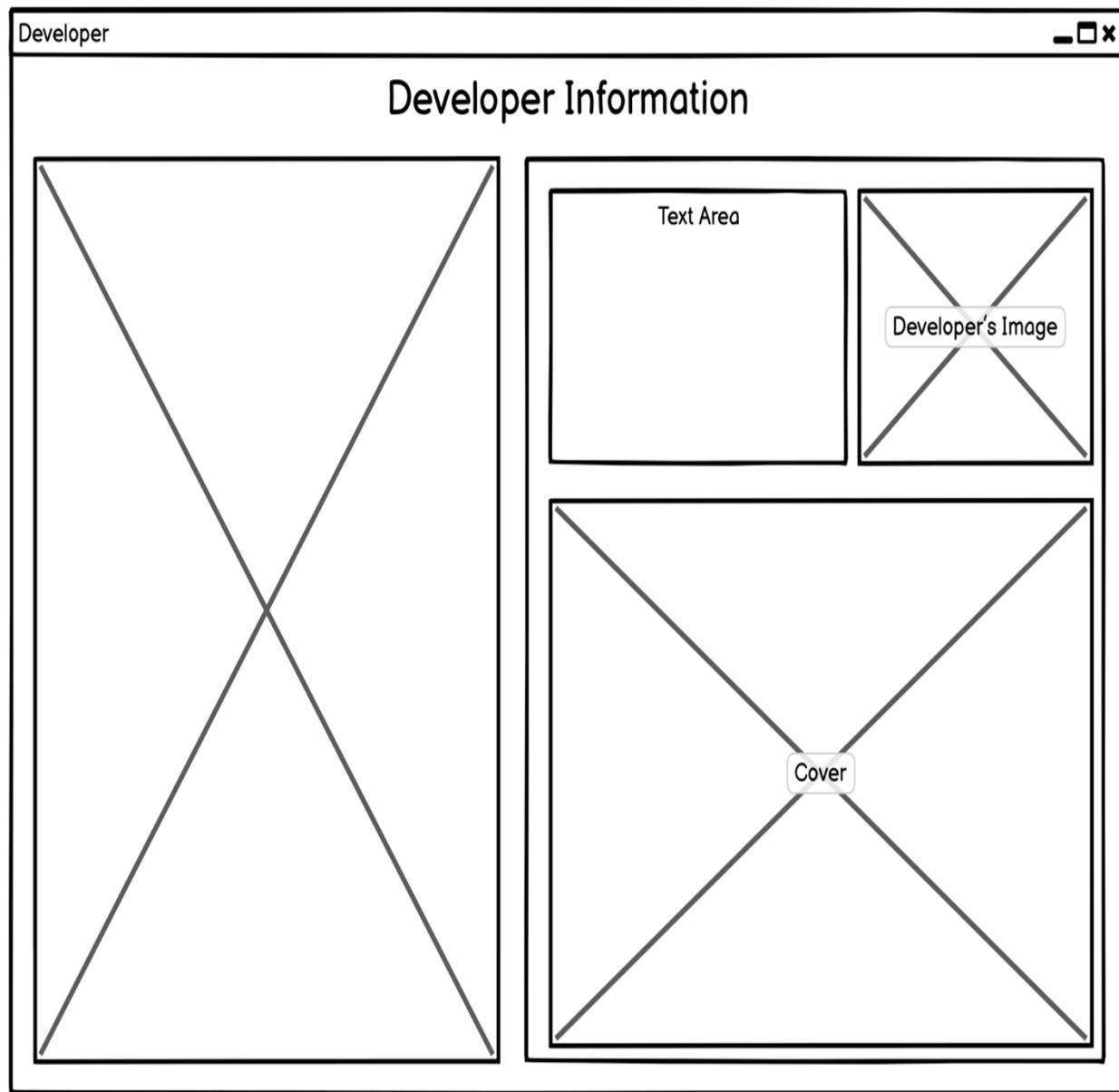


Figure 32: Wireframe - Developer window

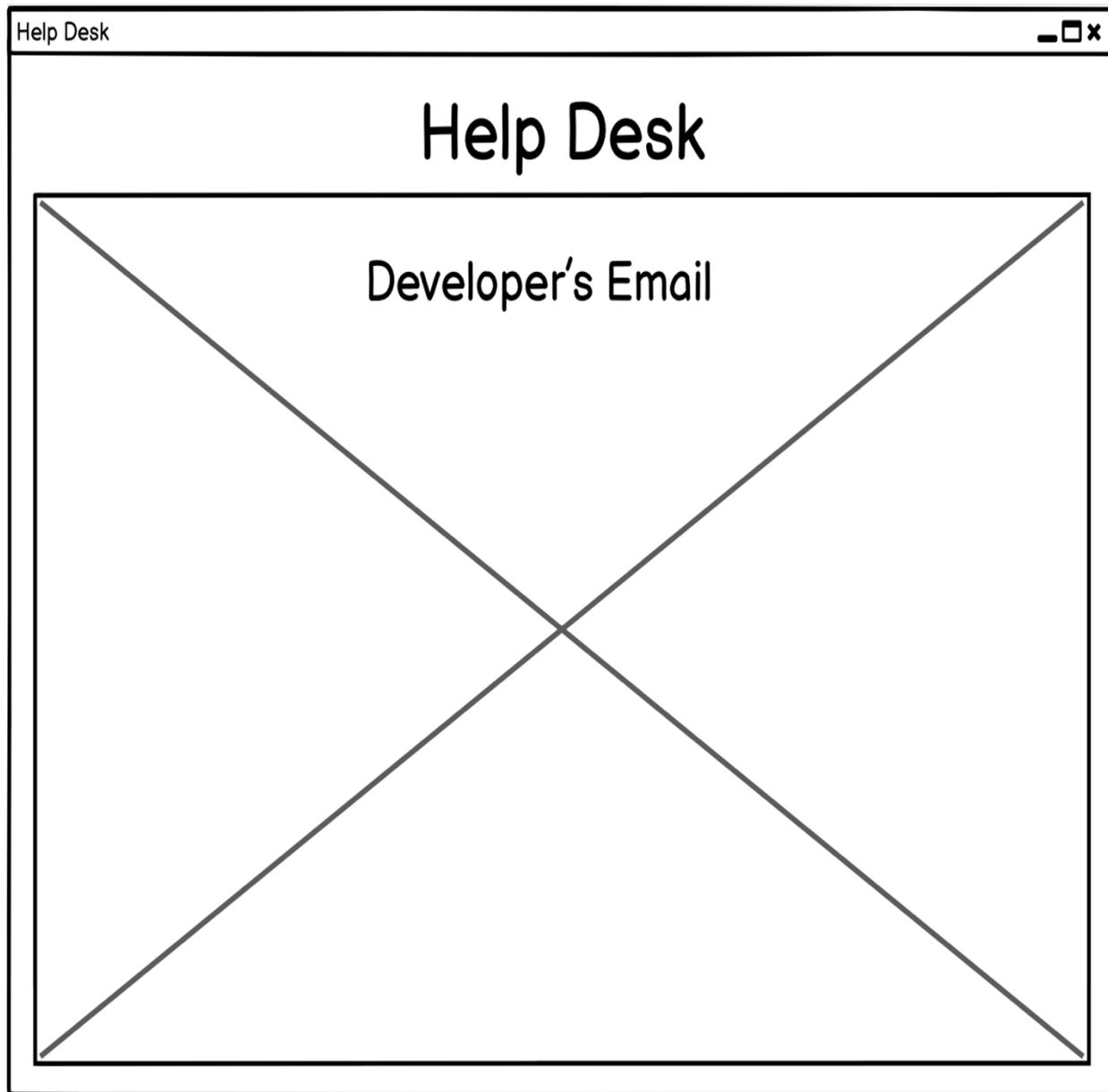


Figure 33: Wireframe - Help Desk Window

**3.8.1.4 User Evaluation:**

Following the completion of the prototype, I had a discussion about the system's many aspects. In order to store student information in a database, the administrator needs have database access. As a result, I utilized the MySQL database included with Xampp because it is the simplest and most widely used database.

**3.8.1.5 Refining a prototype:**

It was all good as required after the client's evaluation. As a result, there is no need for additional refinement. As a result, it became a final prototype, which is suitable for further development.

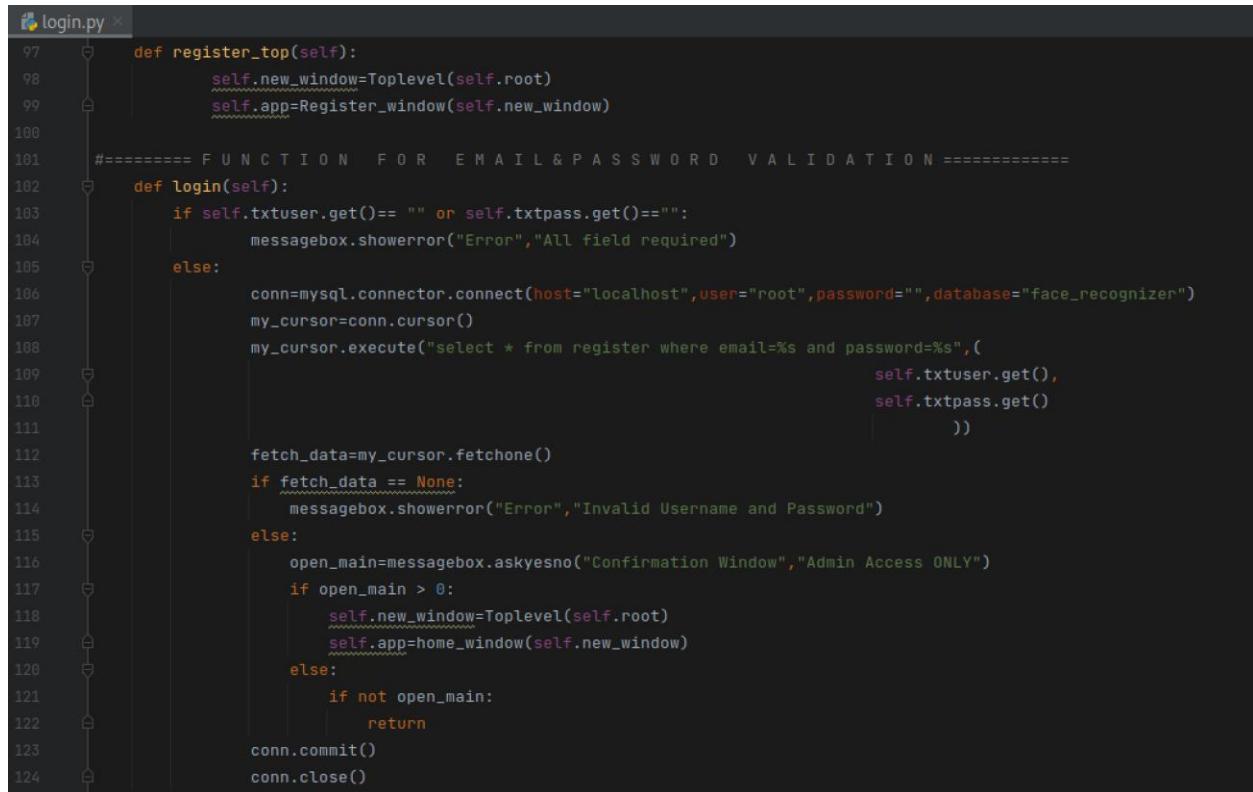
**3.8.1.6 Implement the features**

The admin position now has all of its capabilities and functions.

### 3.8.2 GUI Design

#### 3.8.2.1 Login Window

This code below is used as function in login window where login data is fetched from the database.



```
login.py
97     def register_top(self):
98         self.new_window=Toplevel(self.root)
99         self.app=Register_window(self.new_window)
100
101     ##### F U N C T I O N   F O R   E M A I L & P A S S W O R D   V A L I D A T I O N #####
102     def login(self):
103         if self.txtuser.get() == "" or self.txtpass.get() == "":
104             messagebox.showerror("Error","All field required")
105         else:
106             conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
107             my_cursor=conn.cursor()
108             my_cursor.execute("select * from register where email=%s and password=%s",
109                             (self.txtuser.get(),
110                              self.txtpass.get()
111                             ))
112             fetch_data=my_cursor.fetchone()
113             if fetch_data == None:
114                 messagebox.showerror("Error","Invalid Username and Password")
115             else:
116                 open_main=messagebox.askyesno("Confirmation Window","Admin Access ONLY")
117                 if open_main > 0:
118                     self.new_window=Toplevel(self.root)
119                     self.app=home_window(self.new_window)
120                 else:
121                     if not open_main:
122                         return
123             conn.commit()
124             conn.close()
```

Figure 34: Code - Login Window

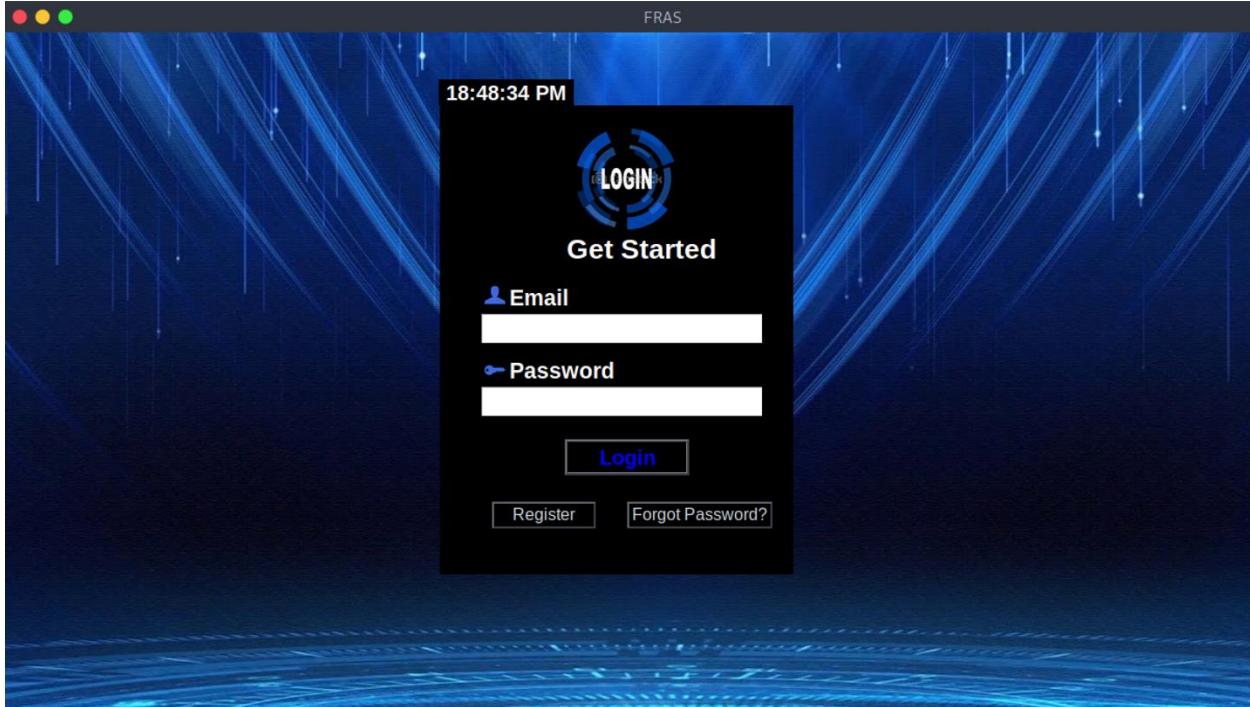


Figure 35: GUI - Login window

### 3.8.2.2 Register Window

This window is designed GUI for registering new admin or teacher in order to login and access the system. The code below is functions used in Register Window.

```

309     #Function==Declaration
310     def register_data(self):
311         if self.var_first_name.get() == "" or self.var_email.get() == "" or self.var_security_qst.get() == "Select" or self.var_contact.get() == "" or self.var_security_ans.get() == "":
312             messagebox.showerror("Error", "All fields are required")
313         elif self.var_pass.get() != self.var_confirm_pass.get():
314             messagebox.showerror("Error", "Unmatched password")
315         elif self.var_check_button.get() == 0:
316             messagebox.showerror("Error", "Must agree terms & conditions")
317         elif not ("@" or ".com") in self.var_email.get():
318             messagebox.showerror("Error occurred", "Invalid email! Please Enter email like <santos@gmail.com>", parent=self.root)
319         elif not ("@" or "!" or "$" or "-" or "." or "#") in self.var_pass.get():
320             messagebox.showerror("Password Error", "Please, Enter strong password like SanTos@123", parent=self.root)
321         else:
322             conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
323             my_cursor=conn.cursor()
324             query="Select * from register where email=%s"
325             value=(self.var_email.get(),)
326             my_cursor.execute(query,value)
327             fetched_data=my_cursor.fetchone()
328             if fetched_data!=None:
329                 messagebox.showerror("Error", "Existing User, Try another email")
330             else:
331                 my_cursor.execute("INSERT INTO register VALUE(%s,%s,%s,%s,%s,%s,%s)",(
332                     self.var_first_name.get(),
333                     self.var_last_name.get(),
334                     self.var_contact.get(),
335                     self.var_email.get(),
336                     self.var_pass.get(),
337                     self.var_confirm_pass.get(),
338                     self.var_security_qst.get(),
339                 ))
340             conn.commit()
341             conn.close()
342             messagebox.showinfo("Success", "Registration successful", parent=self.root)
343             self.var_email.set("")
344             self.var_pass.set("")
345             self.var_confirm_pass.set("")
346             self.var_security_qst.set("Select")
347             self.var_contact.set("")
348             self.var_first_name.set("")
349             self.var_last_name.set("")
350             self.var_email.focus()
351     
```

Figure 36: Code - Register window i

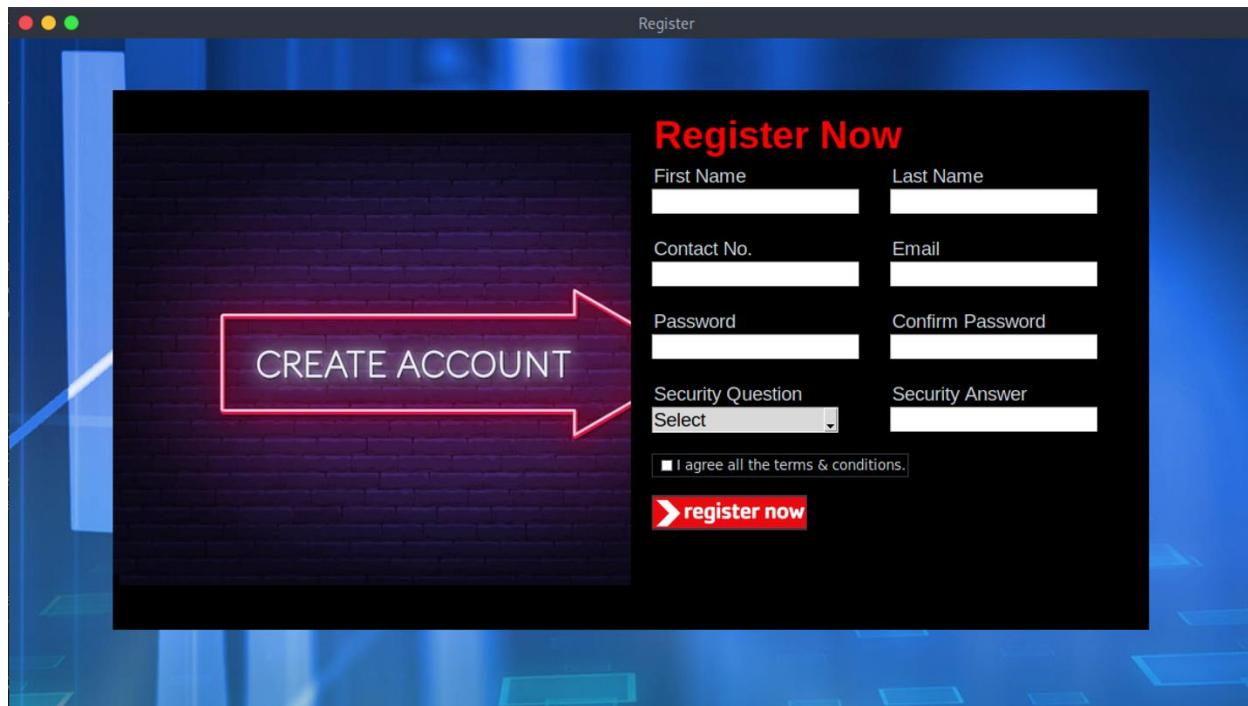


Figure 37: GUI - Register window

### 3.8.2.3 Forgot Password Window

This window is the password recovery window where respective must enter his/her email then click on forget password to access this window. Following code is the functional part of this window.

```
#===== R E S E T   P A S S W O R D   F U N C T I O N =====
def reset_pass(self):
    if self.combo_security.get()=="Select":
        messagebox.showerror("Error","Select the security question",parent=self.root2)
    elif self.security_ans_entry.get()=="":
        messagebox.showerror("Error","Please enter the answer",parent=self.root2)
    elif self.new_password_entry.get()=="":
        messagebox.showerror("Error","Please Enter new password",parent=self.root2)
    else:
        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
        my_cursor=conn.cursor()
        query=("SELECT * FROM register WHERE email=%s AND SecurityQuestion=%s AND SecurityAnswer=%s")
        value=(self.txtUser.get(),self.combo_security.get(),self.security_ans_entry.get())
        my_cursor.execute(query,value)
        fetch_data=my_cursor.fetchone()
        if fetch_data==None:
            messagebox.showerror("Error","Please enter correct answer",parent=self.root2)
        else:
            query=("UPDATE register SET password=%s WHERE email=%s")
            value=(self.new_password_entry.get(),self.txtuser.get())
            my_cursor.execute(query,value)
            messagebox.showinfo("Info","New password set, login now",parent=self.root2)
        conn.commit()
        conn.close()
        self.root2.destroy()
```

Figure 38: Code - Register window

```
#===== F O R G E T   P A S S W O R D   F U N C T I O N =====
def forgot_password_window(self):
    if self.txtuser.get()=="":
        messagebox.showerror("Alert","Enter email section to reset password")
    else:
        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
        my_cursor=conn.cursor()
        query=("SELECT * FROM register WHERE email=%s")
        value=(self.txtuser.get(),)
        my_cursor.execute(query,value)
        fetch_data=my_cursor.fetchone()
        if fetch_data==None:
            messagebox.showerror("Error","Email or Username not Found")
        else:
            conn.close()
            self.root2=Toplevel()
            self.root2.title("Forgot Password")
            self.root2.geometry("400x420+400+120")
```

Figure 39: Code - Register window

### 3.8.2.4 Home Page Window

This is the main window of this system which can be accessible only after login authorization.

Illustrated code is the functions of this window followed by its GUI.

```
===== Function Buttons =====
#Student Window Redirect
def student_details(self):
    self.new_window=Toplevel(self.root)
    self.app=Student(self.new_window)

#Train Windows Redirect
def train_data(self):
    self.new_window=Toplevel(self.root)
    self.app=Train(self.new_window)

#Face_recognition Windows Redirect
def face_data(self):
    self.new_window=Toplevel(self.root)
    self.app=face_recognition(self.new_window)

#attendance window redirect
def attendance_sheet(self):
    self.new_window=Toplevel(self.root)
    self.app=attendance(self.new_window)

#Developer Window redirect
def developer(self):
    self.new_window=Toplevel(self.root)
    self.app=developer(self.new_window)
```

Figure 40: Code - Home window (i)

```

#Help Desk windows redirect
def help_desk(self):
    self.new_window=Toplevel(self.root)
    self.app=help(self.new_window)

#Exit on click
def iExit(self):
    self.iExit=tkinter.messagebox.askyesno("Face Recognition","Do you want to QUIT this project?",parent=self.root)
    if self.iExit >0:
        self.root.destroy()
    else:
        return

#Photos
def open_img(self):
    os.startfile("/home/hype/Downloads/FYP-Development/VS-Code/Data")

```

Figure 41: Code - Home Window ii

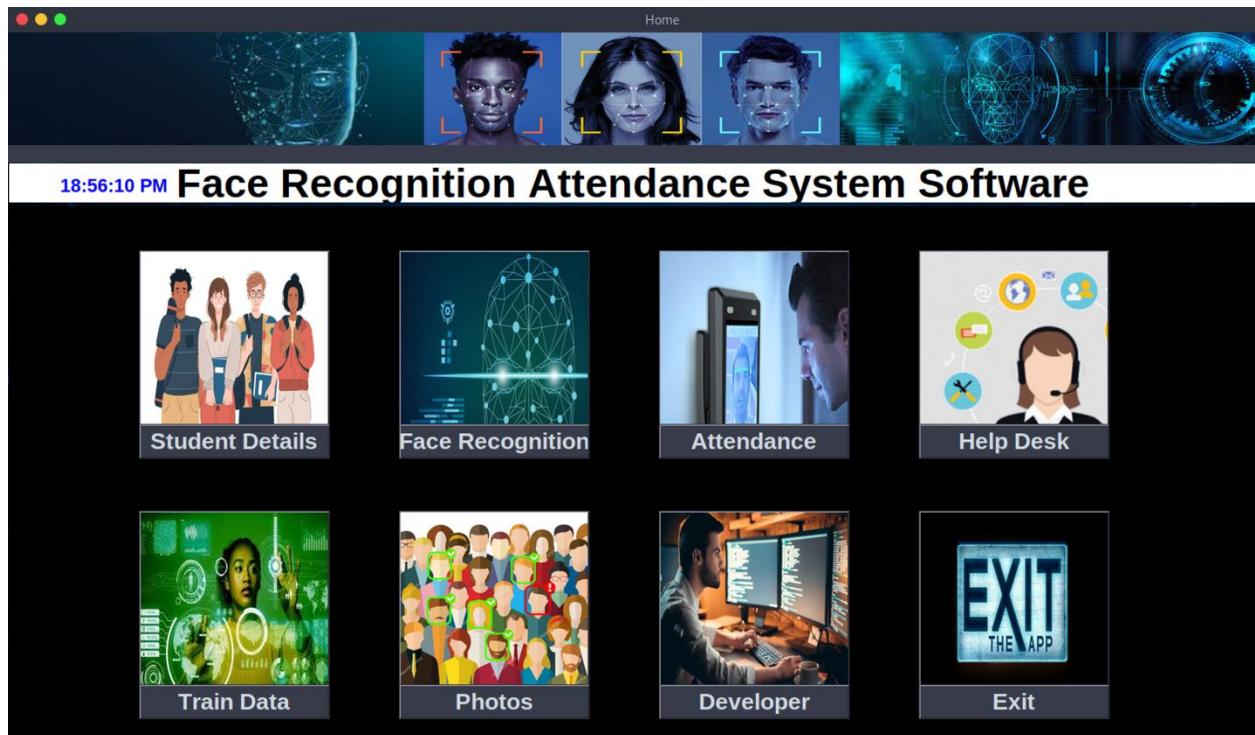


Figure 42: GUI - Home window

### **3.8.2.5 Student Detail Window**

This is window where admin or teacher can enrol new student into class, following with every detail entry of student. Dataset of student detail stored through this window into the database.

Illustrated code is chunk of function of this window followed by its GUI.

Figure 43: Code - Student detail window in

```
#=====Fetch data=====
def fetch_data(self):
    conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
    my_cursor=conn.cursor()
    my_cursor.execute("SELECT * FROM student")
    my_data=my_cursor.fetchall()

    if len(my_data) != 0:
        self.student_table.delete(*self.student_table.get_children())
        for i in my_data:
            self.student_table.insert("",END,values=i)
        conn.commit()

    conn.close()
```

Figure 44: Code - Student detail iii

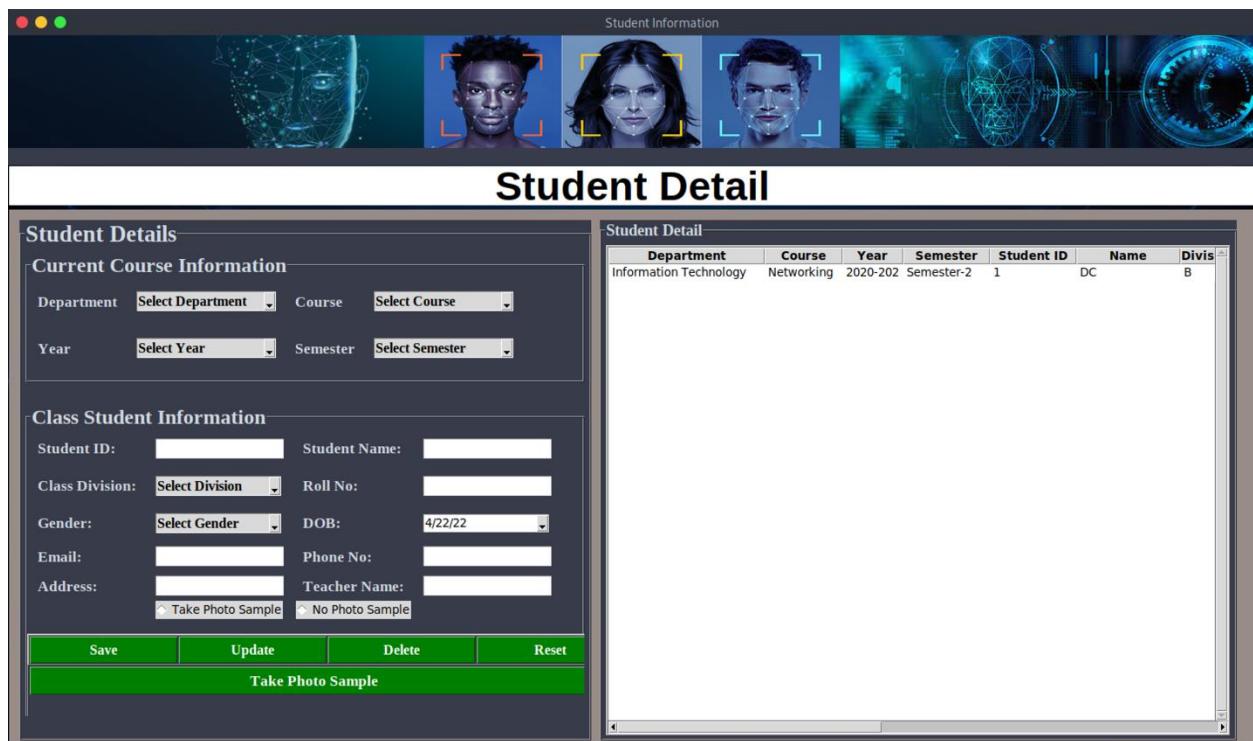


Figure 45: GUI - Student detail

### 3.8.2.6 Train Data Window

This window act as dataset trainer where a single click on its button performs training of dataset through the HAAR-Cascade algorithm. Face is cropped and turned into grayscale in this section.

Chunk of code is illustrated below with its GUI.

```
#=====T R A I N   C L A S S I F I E R   F U N C T I O N=====
def train_classifier(self):
    data_dir=(r"/home/hype/Downloads/FYP-Development/VS-Code/Data")
    path=[os.path.join(data_dir,file) for file in os.listdir(data_dir)]

    faces=[]
    ids=[]
    for image in path:
        img=Image.open(image).convert('L')      #Gray Scale image
        imageNp=np.array(img, 'uint8') #uint8 is datatype
        id=int(os.path.split(image)[1].split('.')[1])

        faces.append(imageNp)
        ids.append(id)
        cv2.imshow("Training Data",imageNp)
        cv2.waitKey(1)==13
    ids=np.array(ids)

#=====train classifier and save=====
clf=cv2.face.LBPHFaceRecognizer_create()
clf.train(faces,ids)
clf.write("/home/hype/Downloads/FYP-Development/VS-Code/Classifier.xml")
cv2.destroyAllWindows()
messagebox.showinfo("Result","Training Dataset Completed",parent=self.root)
```

Figure 46: Code - Train data

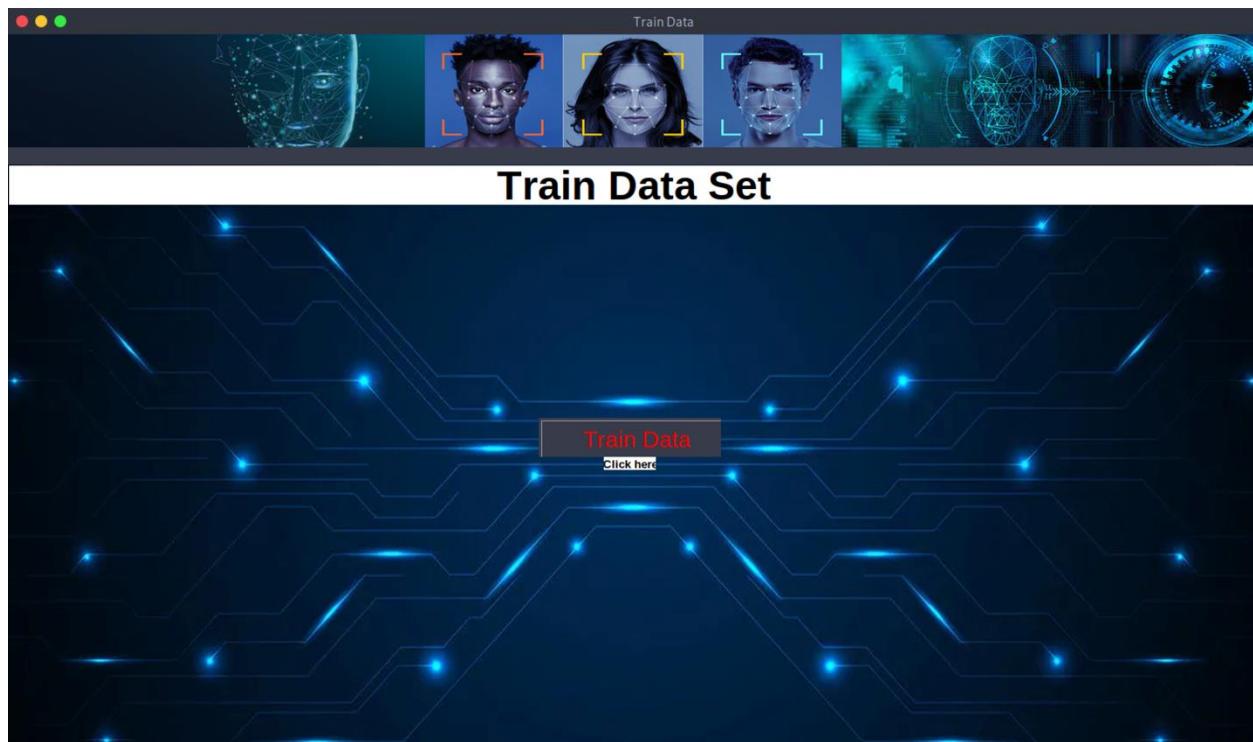


Figure 47: GUI - Train data

### 3.8.2.7 Face Recognition Window

This is one of the core windows of this system where a single click on button launch camera and live video capture and detect faces and similarly generate a csv file with attendance of recognized faces. Illustrated code is the chunk of its function followed by its GUI.

```
#=====F A C E   R E C O G N I T I O N   F U N C T I O N =====
def face_recog(self):
    def draw_boundary(img,classifier,scaleFactor,minNeighbors,color,text,clf):
        gray_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        features=classifier.detectMultiScale(gray_image,scaleFactor,minNeighbors)

        coord=[] #For rectangle over face

        for (x,y,w,h) in features:
            cv2.rectangle(img,(x,y),(x+w+20,y+h+20),(0,255,0),1)
            id,predict=clf.predict(gray_image[y:y+h+20,x:x+w+20])

            conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
            my_cursor=conn.cursor()

            my_cursor.execute("SELECT StudentID FROM student WHERE StudentID=" + str(id))
            i = my_cursor.fetchone()[0]
            i = str(i)

            my_cursor.execute("SELECT Name FROM student WHERE StudentID="+str(id))
            n = my_cursor.fetchone()[0]
            #n=my_cursor.fetchall()
            n=str(n)
            #n="" .join(n)
            #n=join(n)

            my_cursor.execute("SELECT Roll_No FROM student WHERE StudentID=" + str(id))
            r = my_cursor.fetchone()[0]
            #r=my_cursor.fetchall()
            r=str(r)
```

Figure 48: Code - Face recognition window i

```

#Image Recognition function
def recognize(img,clf,faceCascade):
    coord=draw_boundary(img,faceCascade,1.1,10,(255,25,255),"Face",clf)
    return img

faceCascade=cv2.CascadeClassifier("/home/hype/Downloads/FYP-Development/VS-Code/haarcascade_frontalface_default.xml")
clf=cv2.face.LBPHFaceRecognizer_create()
clf.read("/home/hype/Downloads/FYP-Development/VS-Code/Classifier.xml")

#Capture Video with index (-1) to open camera// tested in camera module
video_cap=cv2.VideoCapture(-1)

while True:
    ret,img=video_cap.read()
    img=recognize(img,clf,faceCascade)
    cv2.imshow("Welcome to FRAS",img)
    #cv2.waitKey(1)
    if cv2.waitKey(1)==13:
        break
video_cap.release()
cv2.destroyAllWindows()

```

Figure 49: Code - Face recognition window ii

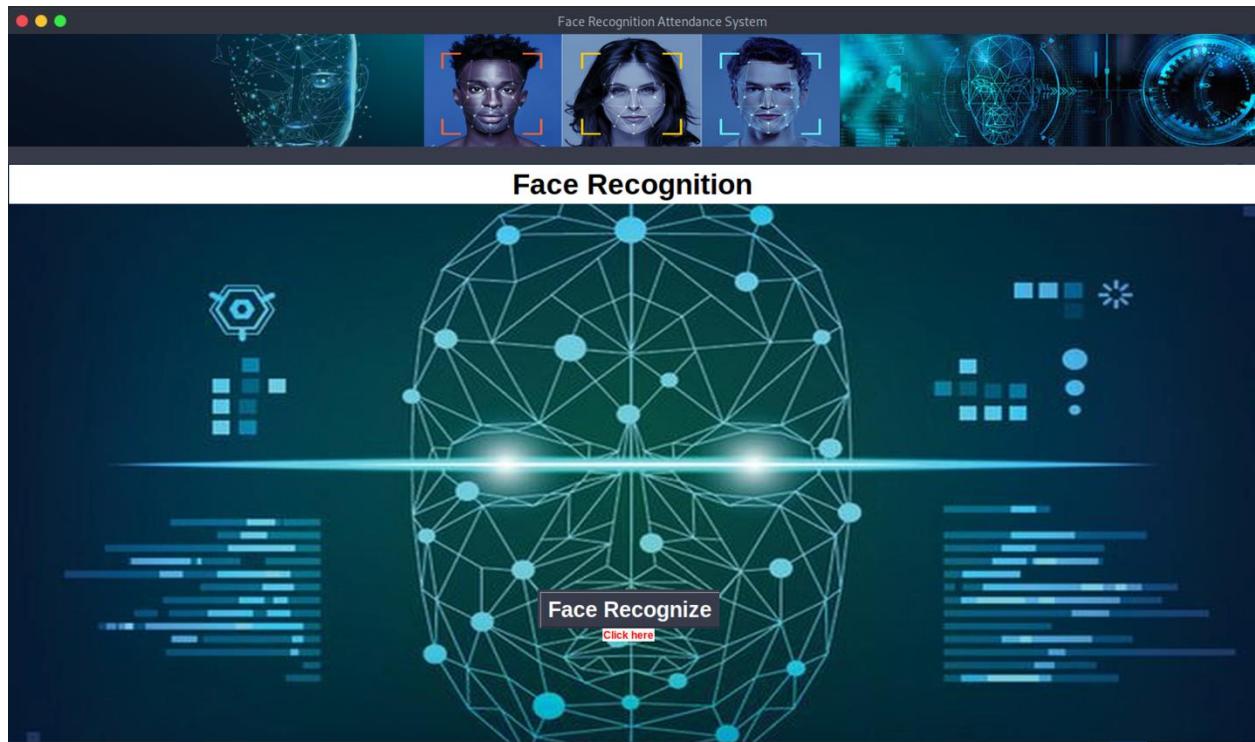


Figure 50: Code - Face Recognition window

### 3.8.2.8 Attendance Window

This window plays vital role in import and export of csv file where admin can modify or view students' attendance. Illustrated code is chunk of functional code with its respective GUI.

```

=====I M P O R T   C S V   F U N C T I O N=====
def importCsv(self):
    global mydata
    mydata.clear()
    fln=filedialog.askopenfilename(initialdir=os.getcwd(),title="Save Csv",filetypes=(("CSV file","*.csv"),("All File","*.*")),parent=self.root)
    with open(fln) as myFile:
        csvRead=csv.reader(myFile,delimiter=",")
        for i in csvRead:
            mydata.append(i)
    self.fetch_data(mydata)

=====E X P O R T   C S V   F U N C T I O N=====
def exportCsv(self):
    try:
        if len(mydata)<1:
            messagebox.showerror("No Data","No Data found to export",parent=self.root)
            return False
        fln=filedialog.asksaveasfilename(initialdir=os.getcwd(),title="Save Csv",filetypes=(("CSV file","*.csv"),("All File","*.*")))
        with open(fln,mode="w",newline="") as myFile:
            exp_write=csv.writer(myFile,delimiter=",")
            for i in mydata:
                exp_write.writerow(i)
            messagebox.showinfo("Data Export","Data is exported to "+os.path.basename(fln)+" successfully")
    except Exception as es:
        messagebox.showerror("Error",f"Due to :{str(es)}",parent=self.root)

```

Figure 51: Code - Attendance window

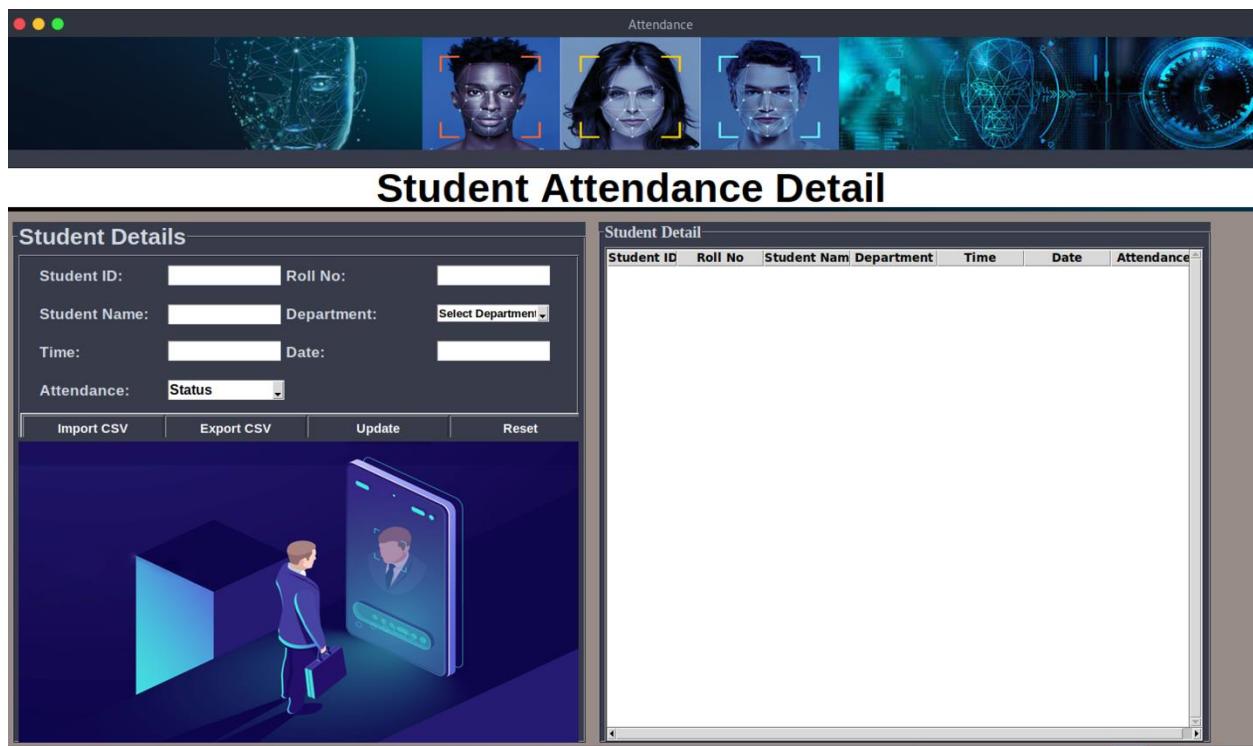


Figure 52: GUI - Attendance window

### 3.8.2.9 Developer Window

This window is the representation of developer's portfolio. Code illustrated below is its GUI code with its GUI included below.

```
from PIL import Image,ImageTk,ImageDraw,ImageFont

class developer:
    #Main Window
    def __init__(self,root):
        self.root=root
        #root.iconbitmap('PATH')
        self.root.geometry("1440x850+0+0")
        self.root.title("Developer")

    #Title Label
        title_lbl=Label(self.root,text=<DEVELOPER/>,font=("arial",35,"bold"),foreground="white",bg="black")
        title_lbl.place(x=0,y=0,width=1400,height=45)

    #Background_img Image1
        background_img=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/developerbg.jpg")
        #cover1=Image.open("Images/cover1.png")
        background_img=background_img.resize((1440,850),Image.ANTIALIAS)
        self.bg_img=ImageTk.PhotoImage(background_img)

        f_lbl=Label(self.root,image=self.bg_img)
        f_lbl.place(x=0,y=42,width=1440, height=850)

    #MAIN FRAME
        main_frame=LabelFrame(f_lbl,relief=RIDGE,bg="black")
        main_frame.place(x=800,y=0,width=600,height=600)
```

Figure 53: Code - Developer window

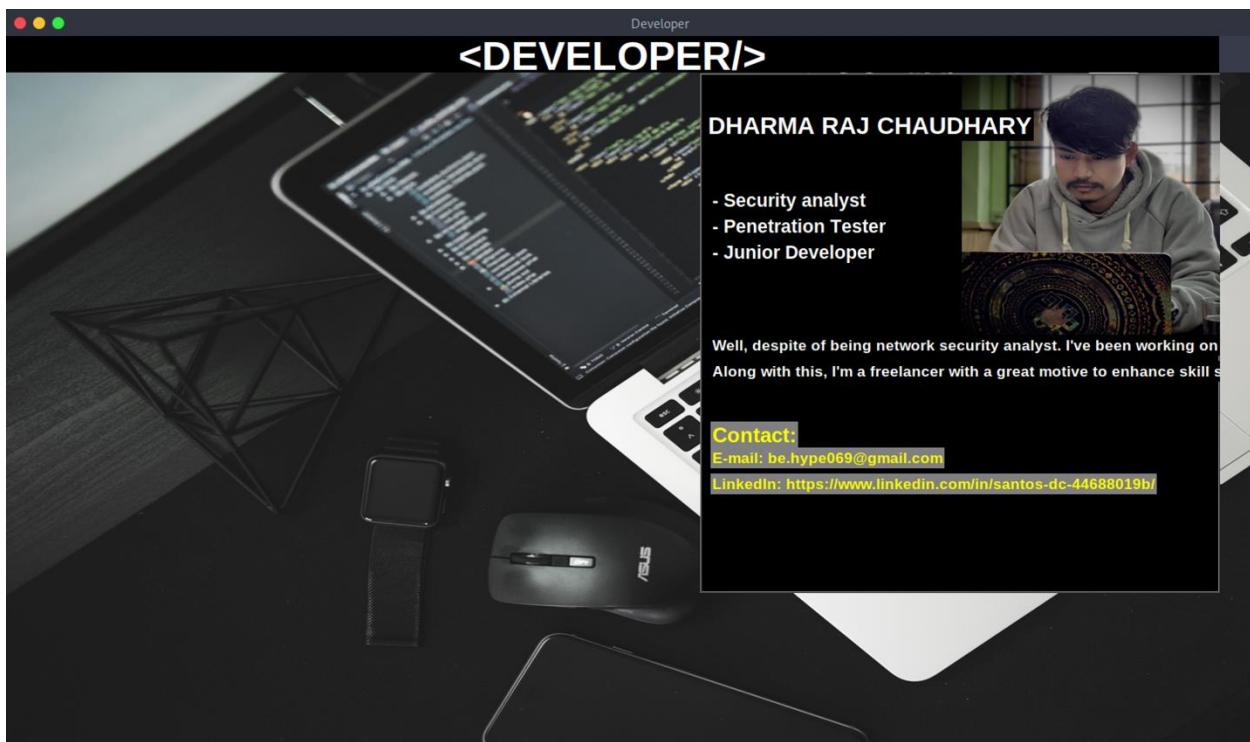


Figure 54: GUI - Developer window

### 3.8.2.10 Help Desk Window

This window is help desk where help lines are presented in here to contact with developer. Illustrated code and its GUI are below:

```
from PIL import Image,ImageTk,ImageDraw,ImageFont
#Main Window
class Help:
    def __init__(self,root):
        self.root=root
        root.iconbitmap('PATH')
        self.root.geometry("1440x850+0+0")
        self.root.title("Help")

    #Title Label
        title_lbl=Label(self.root,text="Help Desk",font=("arial",35,"bold"),foreground="white",bg="Black")
        title_lbl.place(x=0,y=0,width=1400,height=45)

    #Background_img Image
        background_img=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/help_desk.png")
        cover1=Image.open("Images/cover1.png")
        background_img=background_img.resize((1440,850),Image.ANTIALIAS)
        self.bg_img=ImageTk.PhotoImage(background_img)

        f_lbl=Label(self.root,image=self.bg_img)
        f_lbl.place(x=0,y=42,width=1440, height=850)

    #MAIN FRAME
        main_frame=LabelFrame(f_lbl,relief=RIDGE,bg="black")
        main_frame.place(x=200,y=540,width=1020,height=600)

    #Help Lines
        help_desk=Label(main_frame,text="For Any Enquiry",font=("arial",30,"bold"),foreground="white",bg="Black")
```

Figure 55: Code - Help Desk window

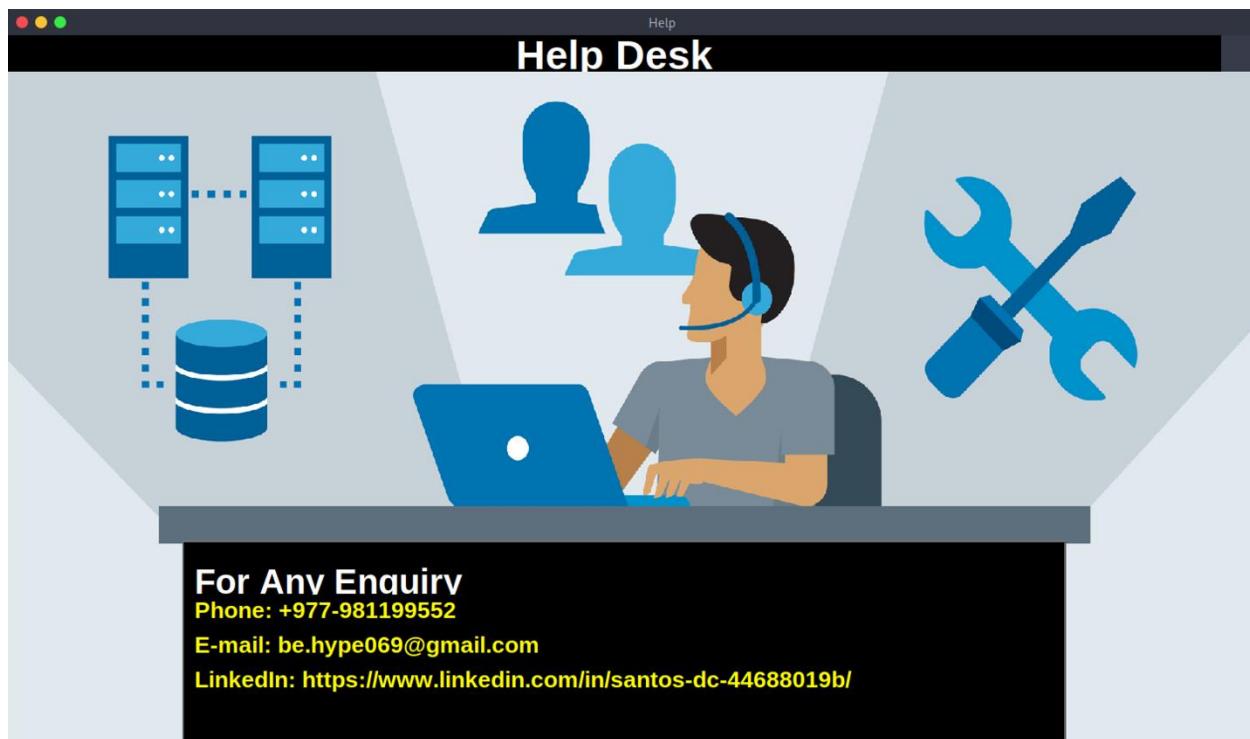


Figure 56: GUI - Help desk window

## Chapter 4: Testing and Analysis

### 4. Testing And Analysis

#### 4.1 Test Plan

White box testing, black box testing, and integration testing are the sorts of testing that will be done. Below are the test plans for testing:

##### 4.1.1 Test plan for white box testing

White box testing is an approach that allows testers to inspect and verify the inner workings of a software system—its code, infrastructure, and integrations with external systems. White box testing is an essential part of automated build processes in a modern Continuous Integration/Continuous Delivery (CI/CD) development pipeline (Imperva, 2021).

The successful test cases will be used in white box testing. Each and every aspect of the unique aspects of the application or system will be properly tested with correct test methods and test cases for successful test cases.

##### 4.1.2 Test plan for black box testing

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues (Imperva, 2021).

The failed test cases will be used in black box testing. First, the error will be displayed, followed by the solution to the mistake.

#### **4.1.3 Test plan for integration testing**

Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated (Hamilton, 2022).

For integration testing, the database integration will be displayed, as well as CRUD operations in the database.

## 4.2 White box testing

### 4.2.1 Successful Test Case for Admin

#### 4.2.1.1 Test 1 – To display the login page

Test	01
Objective	To display the login window
Action	The command ‘login.py’ was run in the IDE.
Expected Result	The login window would be displayed.
Actual Result	The login window is displayed.
Conclusion	Test Successful.

Table 4: Test 1- To display the login page

The screenshot shows the VS Code interface with the 'login.py' file open in the editor. The code implements a registration function with validation logic for fields like first name, email, password, and terms & conditions. It also handles database connection and insertion. Below the editor, the terminal shows the command '/home/hype/Downloads/FYP-Development/venv/bin/python /home/hype/Downloads/FYP-Development/VS-Code/login.py' being run, and the output indicates a successful execution with an exit code of 0.

```

#Function==Declaration
def register_data(self):
    if self.var_first_name.get() == "" or self.var_email.get() == "" or self.var_security_qst.get() == "Select" or self.var_contact.get() == "":
        messagebox.showerror("Error", "All fields are required")
    elif self.var_pass.get() != self.var_confirm_pass.get():
        messagebox.showerror("Error", "Unmatched password")
    elif self.var_check_button.get() == 0:
        messagebox.showerror("Error", "Must agree terms & conditions")
    elif not ("@" or "!" or "$" or "-" or "#") in self.var_email.get():
        messagebox.showerror("Error occurred", "Invalid email! Please Enter email like <santos@gmail.com>", parent=self.root)
    elif not ("@" or "!" or "$" or "-" or "#") in self.var_pass.get():
        messagebox.showerror("Password Error", "Please, Enter strong password like Santos@123", parent=self.root)
    else:
        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
        my_cursor=conn.cursor()
        query="Select * from register where email=%s"
        value=(self.var_email.get(),)
        my_cursor.execute(query,value)
        fetched_data=my_cursor.fetchone()
        if fetched_data==None:
            messagebox.showerror("Error", "Existing User, Try another email")
        else:
            my_cursor.execute("INSERT INTO register VALUE(%s,%s,%s,%s,%s,%s,%s)",(self.var_email.get(),self.var_pass.get(),self.var_security_qst.get(),self.var_contact.get(),self.var_check_button.get(),self.var_first_name.get(),self.var_last_name.get()))
            conn.commit()
            conn.close()

```

Figure 57: Test 1 (i)

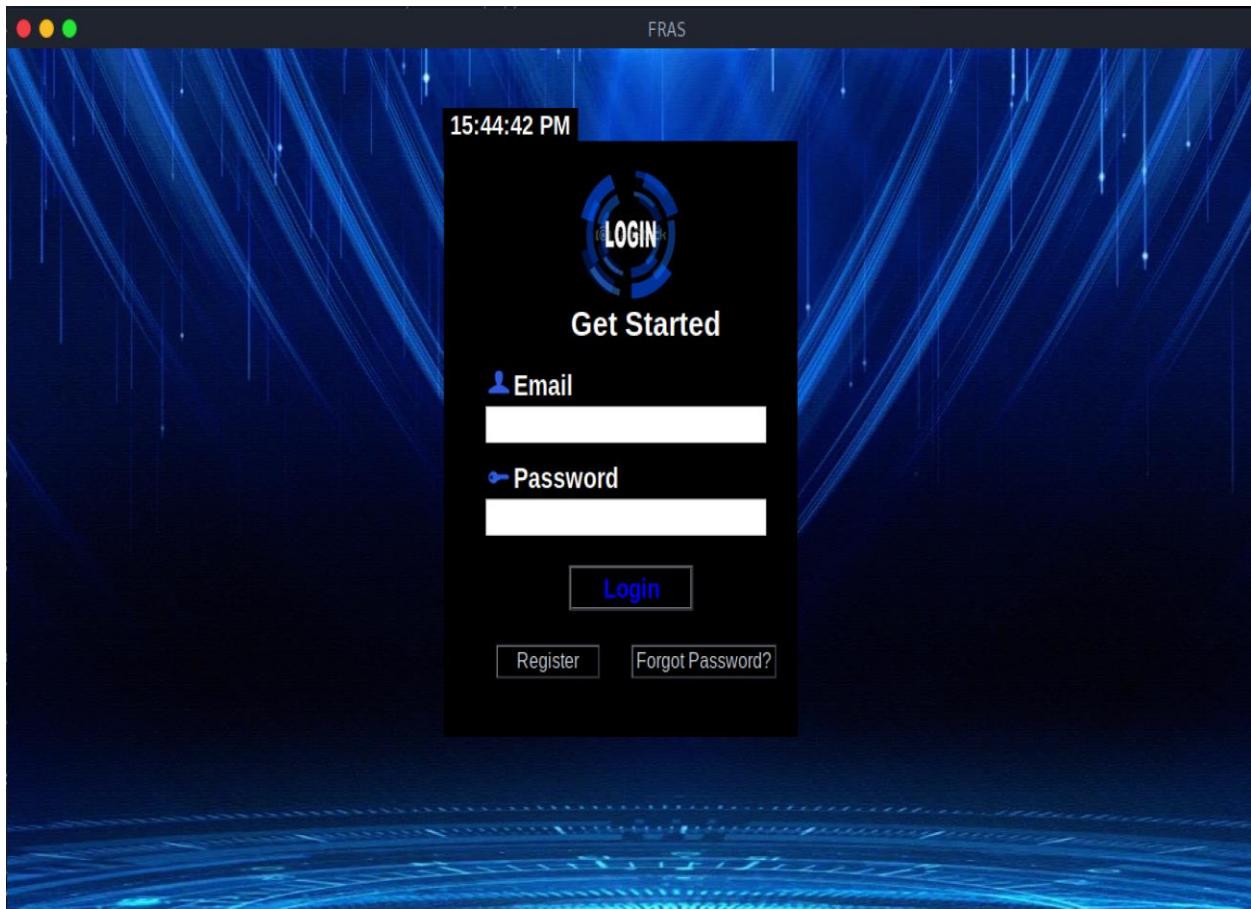


Figure 58: Test 1 (ii)

#### 4.2.1.2 Test 02: To login to the system

Test	02
Objective	To login to the system by entering the email and password.
Action	The admin does not have to register, he/she can directly login to the system entering the email and password. For email field “santos.dc@gmail.com” In password filed “Dc@123” was entered. The “login” button was then clicked.
Expected Result	Admin home window would be displayed.
Actual Result	Admin home window is displayed
Conclusion	Test successful.

Table 5:Test 02: To login to the system

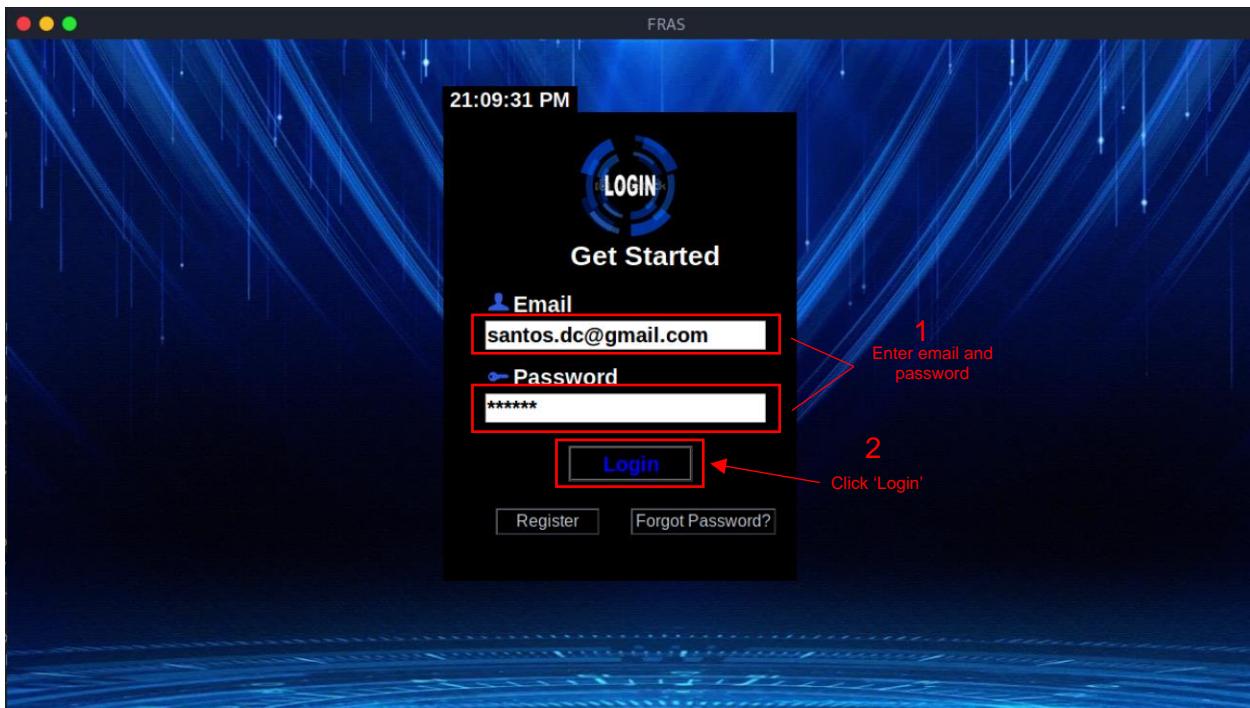


Figure 59: Test 2 (i)

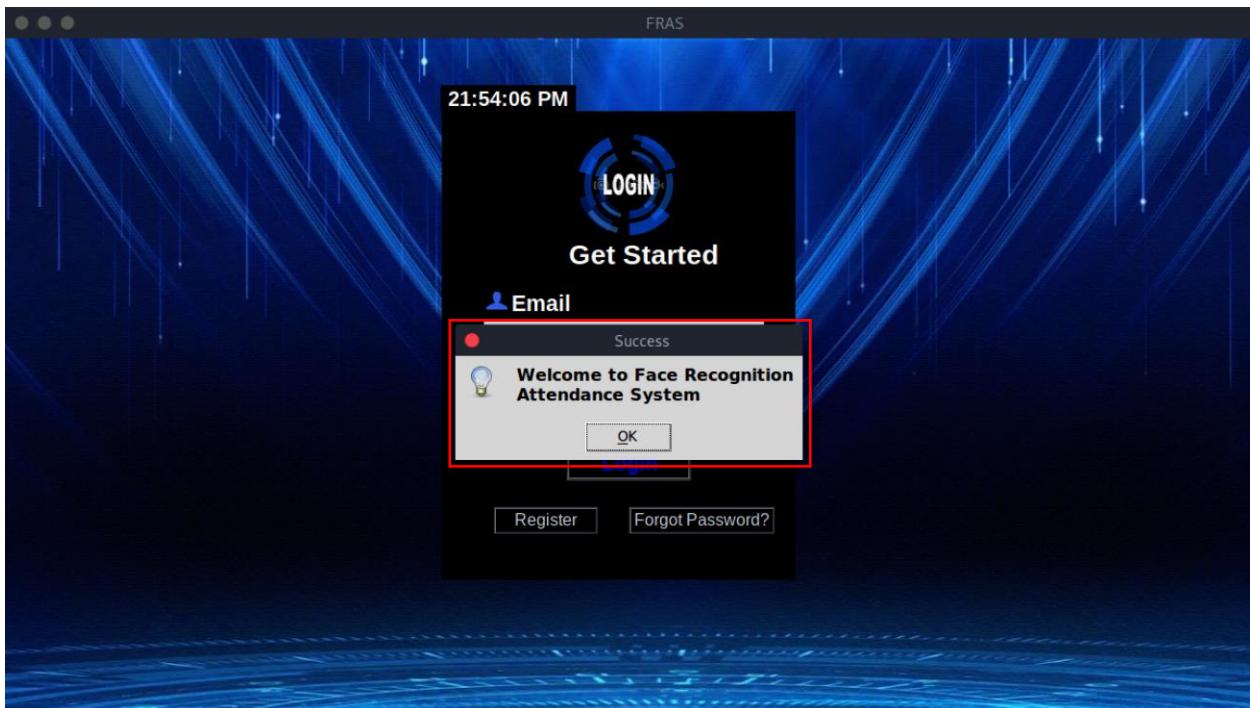


Figure 60: Test 2 (ii)

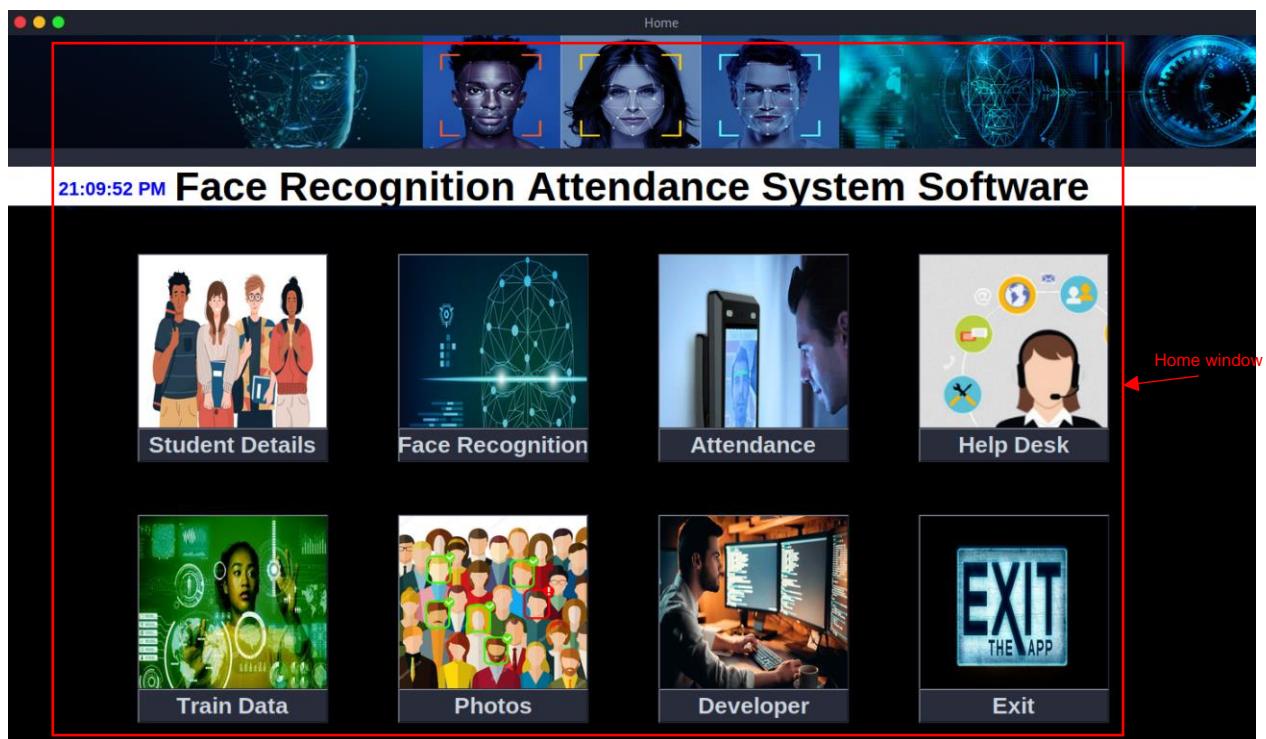


Figure 61: Test 2 (iii)

#### 4.2.1.3 Test 03: To Register New Teacher

Test	03
Objective	To register new teacher who will have access to system
Action	Navigated to 'Register' window in 'Login' window
Expected Result	A register form would be displayed and new registration would be successful.
Actual Result	A register form is displayed and new registration is successful.
Conclusion	Test Successful.

Table 6: Test 03: To Register New Teacher

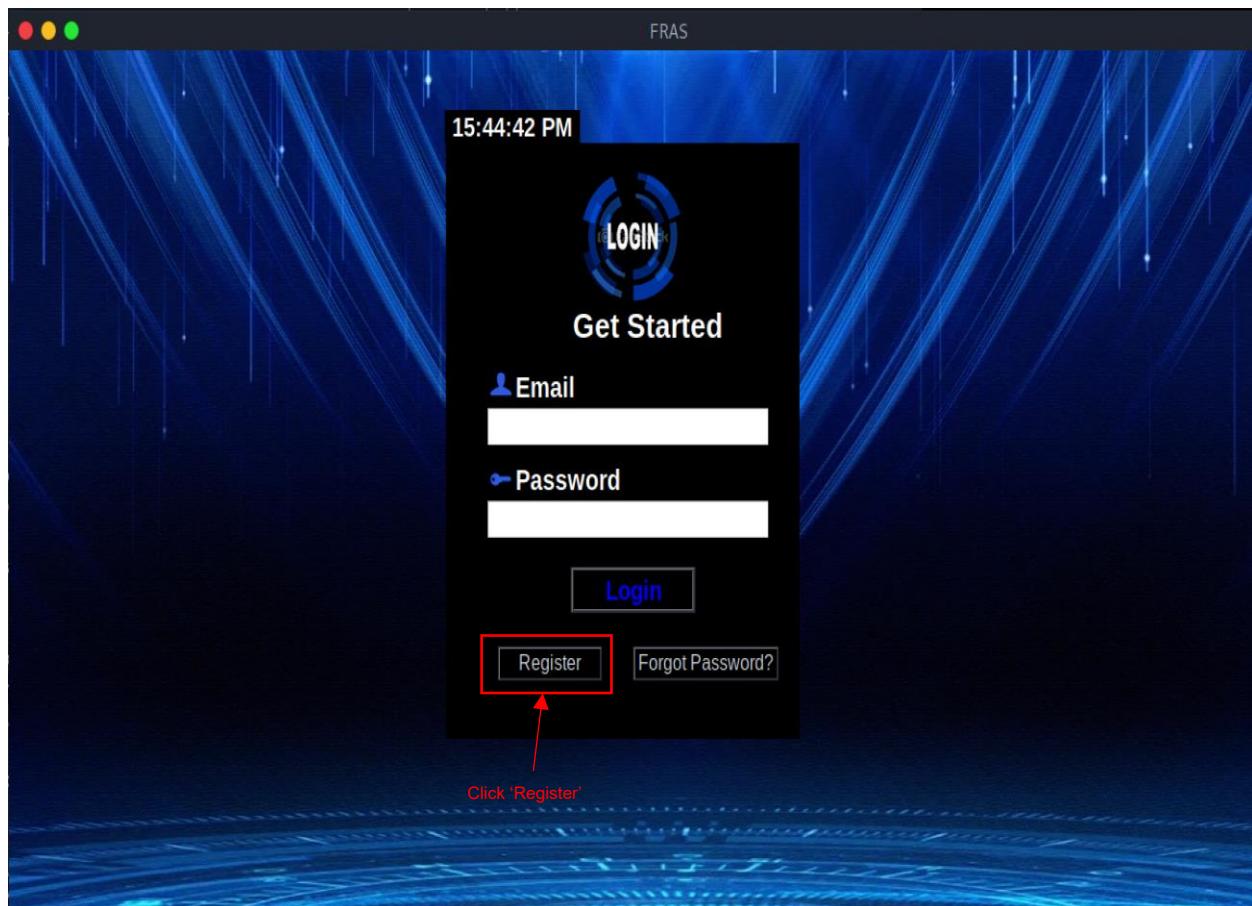


Figure 62:Test 03 (i)

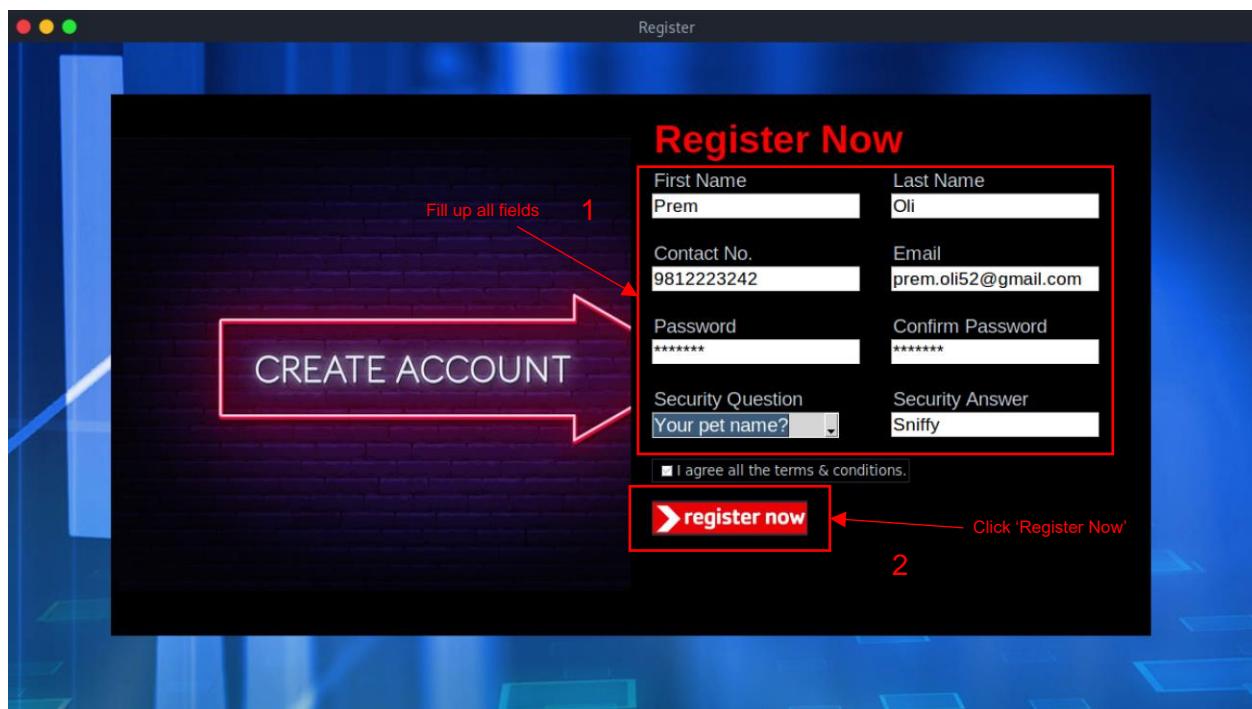


Figure 63: Test 03 (ii)

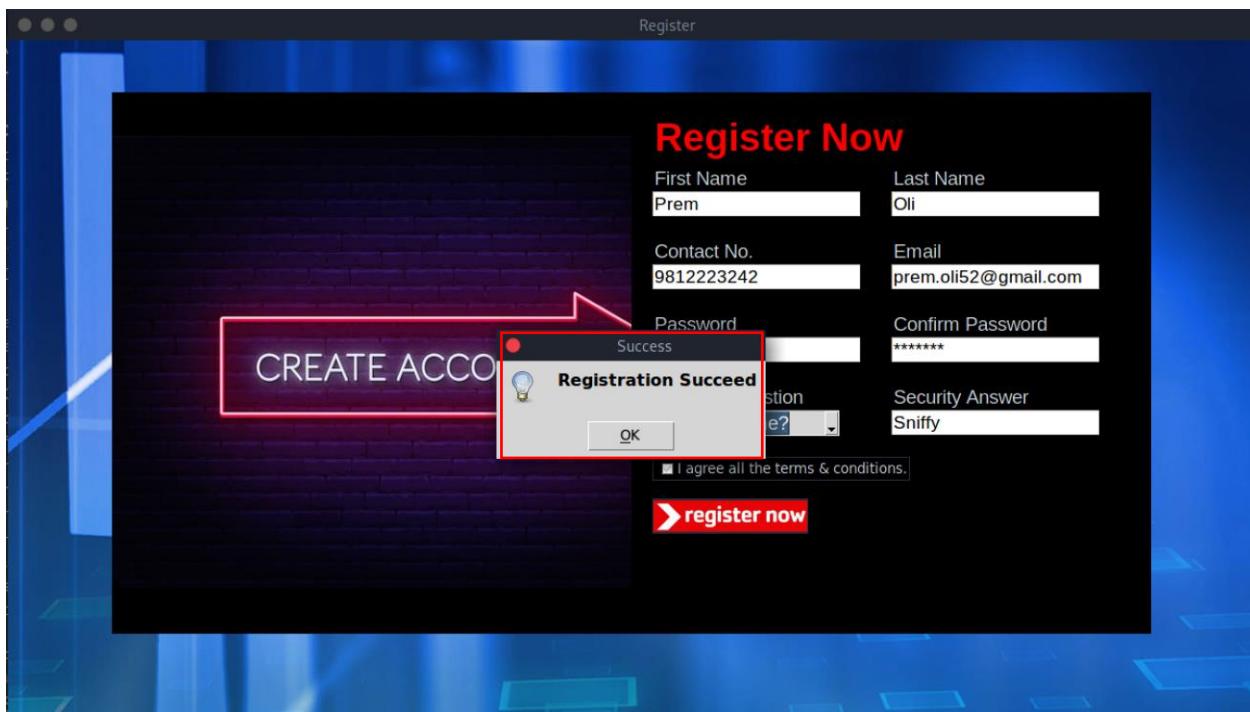


Figure 64: Test 03 (iii)

The screenshot shows the "register" table in the "face\_recognizer" database via phpMyAdmin. The table has columns: First Name, Last Name, Contact No., Email, Password, ConfirmPassword, SecurityQuestion, and SecurityAnswer. Two rows of data are visible, both highlighted with a red box. The first row corresponds to the registration details shown in Figure 64. The second row shows data for "Santosh Chaudhary" with email "santos.dc@gmail.com" and security answers "SanTos@12" and "Parasi".

	First Name	Last Name	Contact No.	Email	Password	ConfirmPassword	SecurityQuestion	SecurityAnswer
1	Prem	Oli	9812223242	prem.ol52@gmail.com	Prem@52	Prem@52	Your pet name?	Sniffy
2	Santosh	Chaudhary	9811995552	santos.dc@gmail.com	SanTos@12	SanTos@12	Your birth place?	Parasi

Figure 65: Test 03 (iv)

#### 4.2.1.4 Test 04: To reset password

Test	04
Objective	To reset password
Action	Filled email on email text area and clicked on ‘Forgot Password’ New window appears, Selected security question and answered it Chose new password and clicked on ‘Reset’ button
Expected Result	Password is supposed to reset
Actual Result	Password is successfully reset.
Conclusion	Test successful.

Table 7: Test 04: To reset password

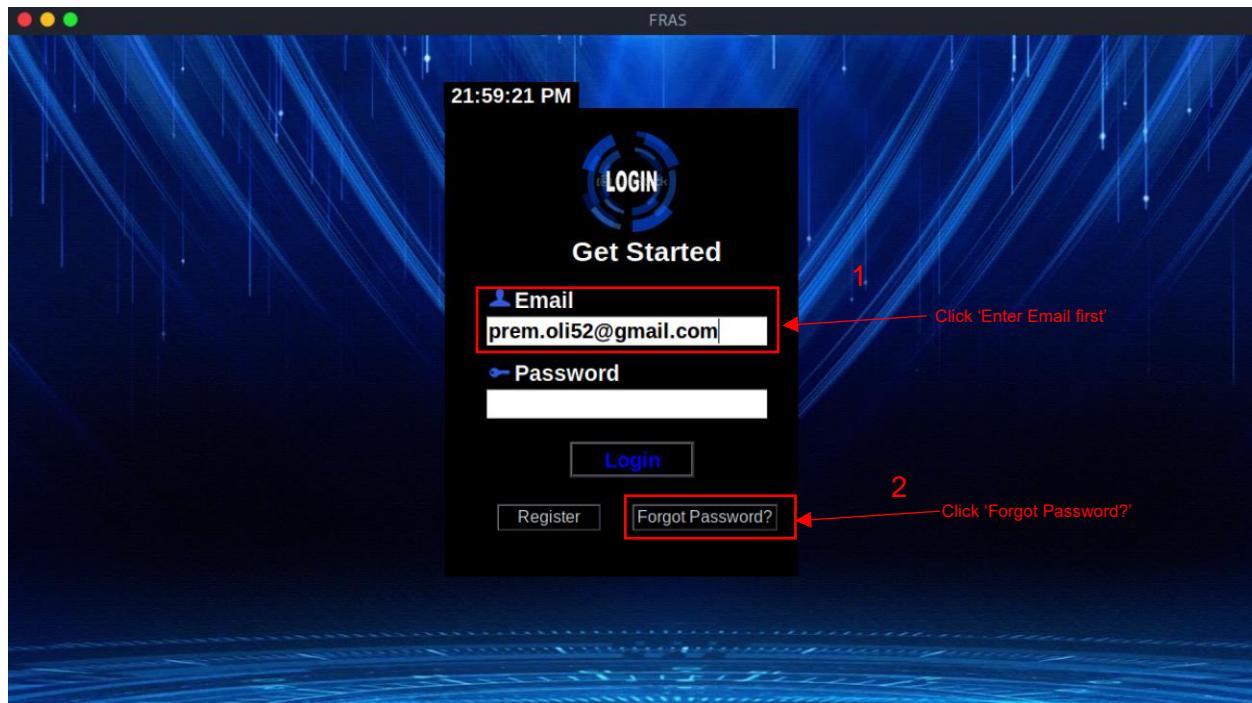


Figure 66: Test 04 (i)

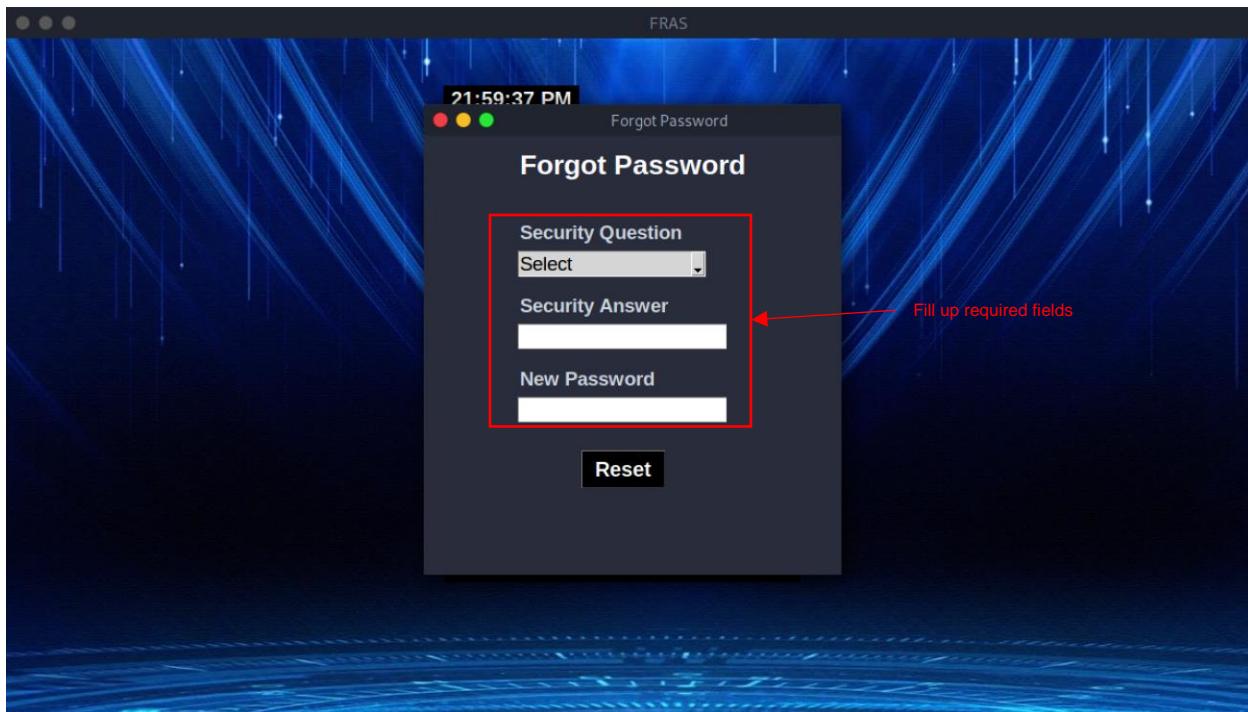


Figure 67: Test 04 (ii)

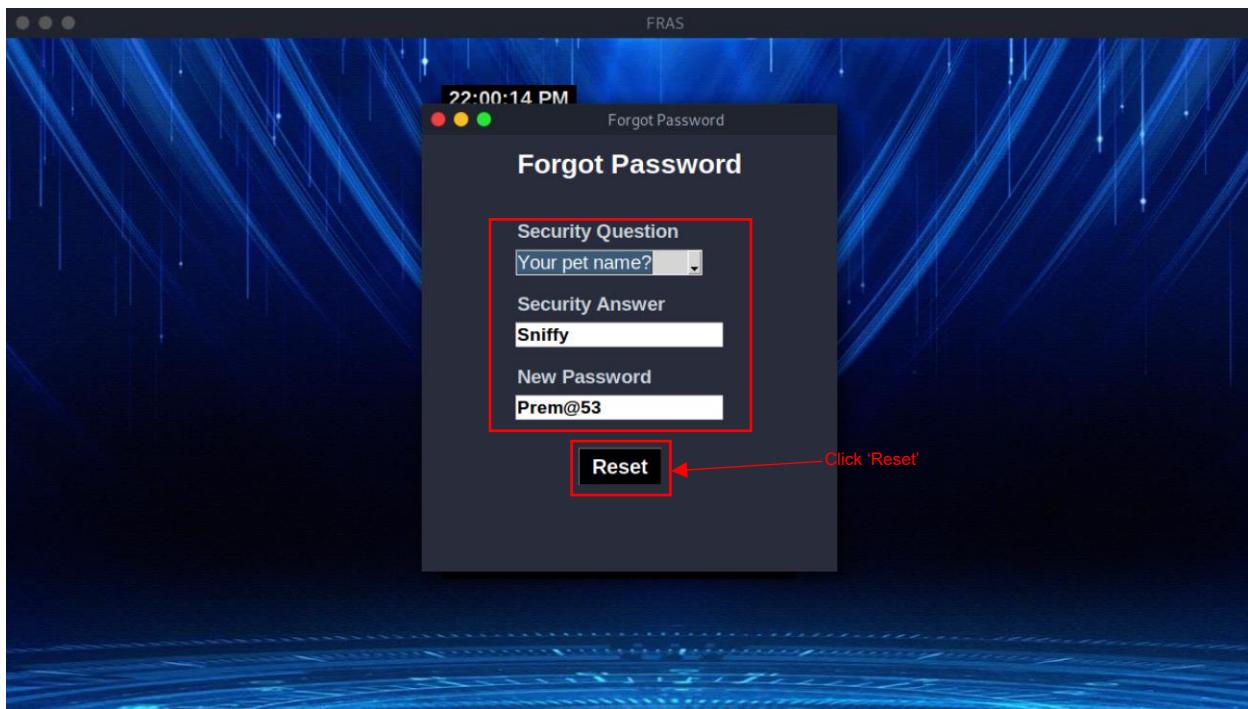


Figure 68: Test 04 (iii)

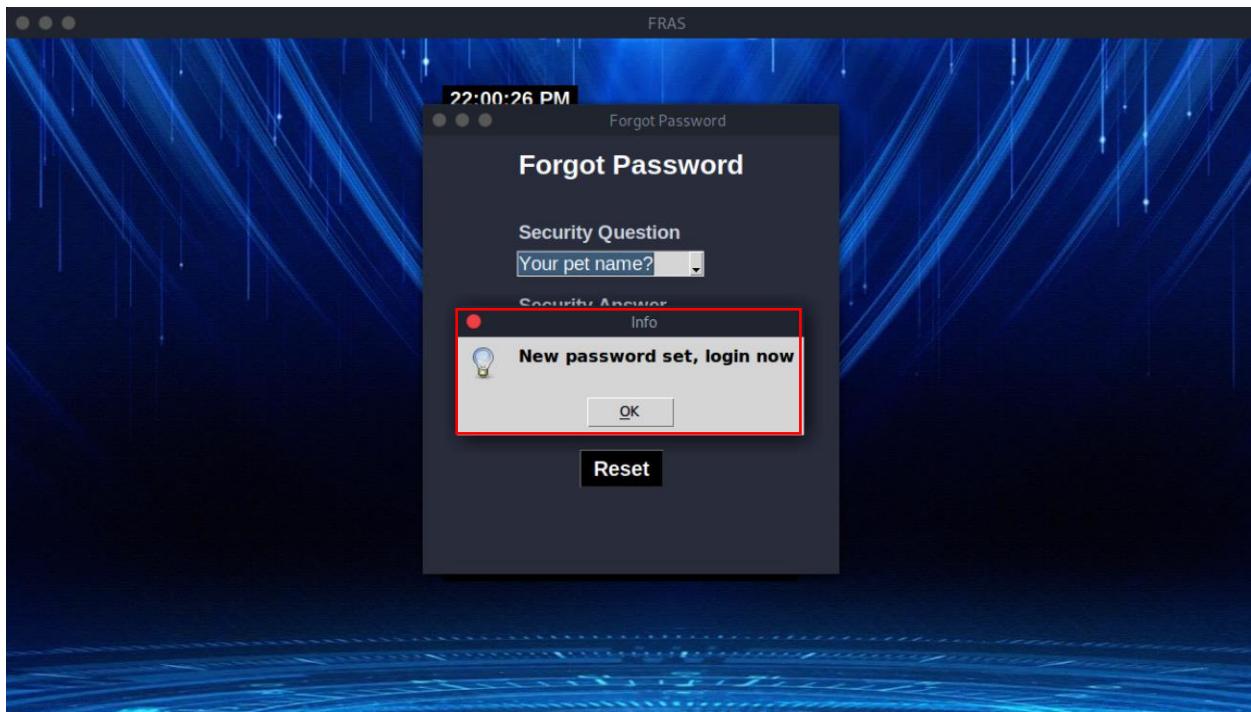


Figure 69: Test 04 (iv)

#### 4.2.1.5 Test 05: To login new registered teacher

Test	05
Objective	To login new registered teacher
Action	Email and Password of new registered teacher is added
Expected Result	System would successfully login new registered teacher
Actual Result	System is successful login new registered teacher
Conclusion	Test successful.

Table 8: Test 05: To login new registered teacher



Figure 70: Test 5 (i)

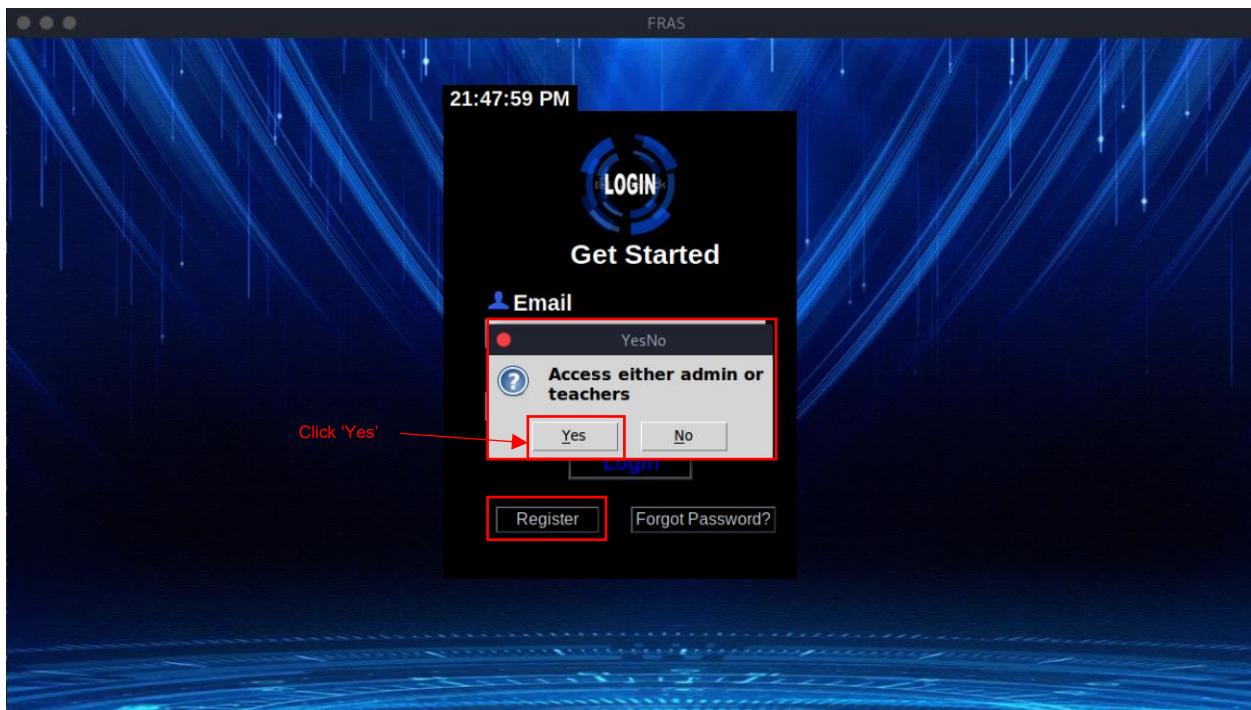


Figure 71: Test 05 (ii)

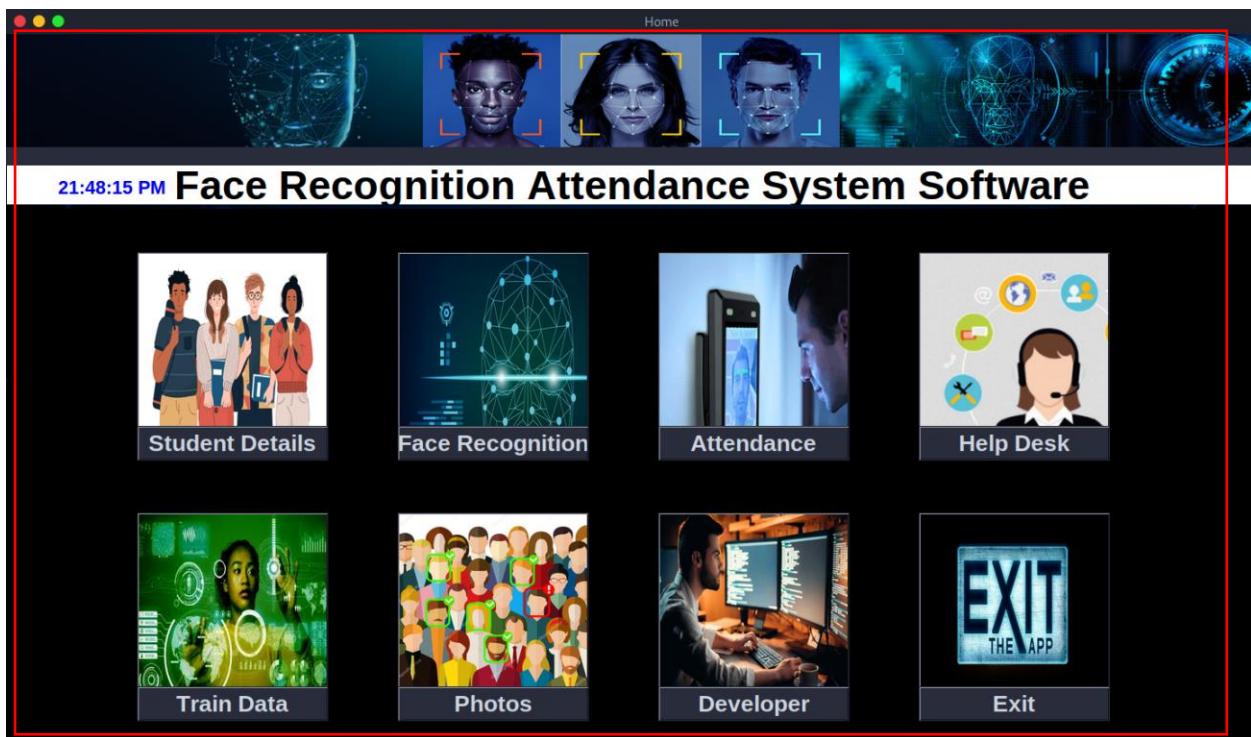


Figure 72: Test 05 (iii)

#### 4.2.1.6 Test 06: Add New Student detail

Test	06
Objective	To add new student
Action	Fill up student detail and take photo and save it to database
Expected Result	System would add new student
Actual Result	System added new student
Conclusion	Test successful.

Table 9: Test 06: Add New Student detail

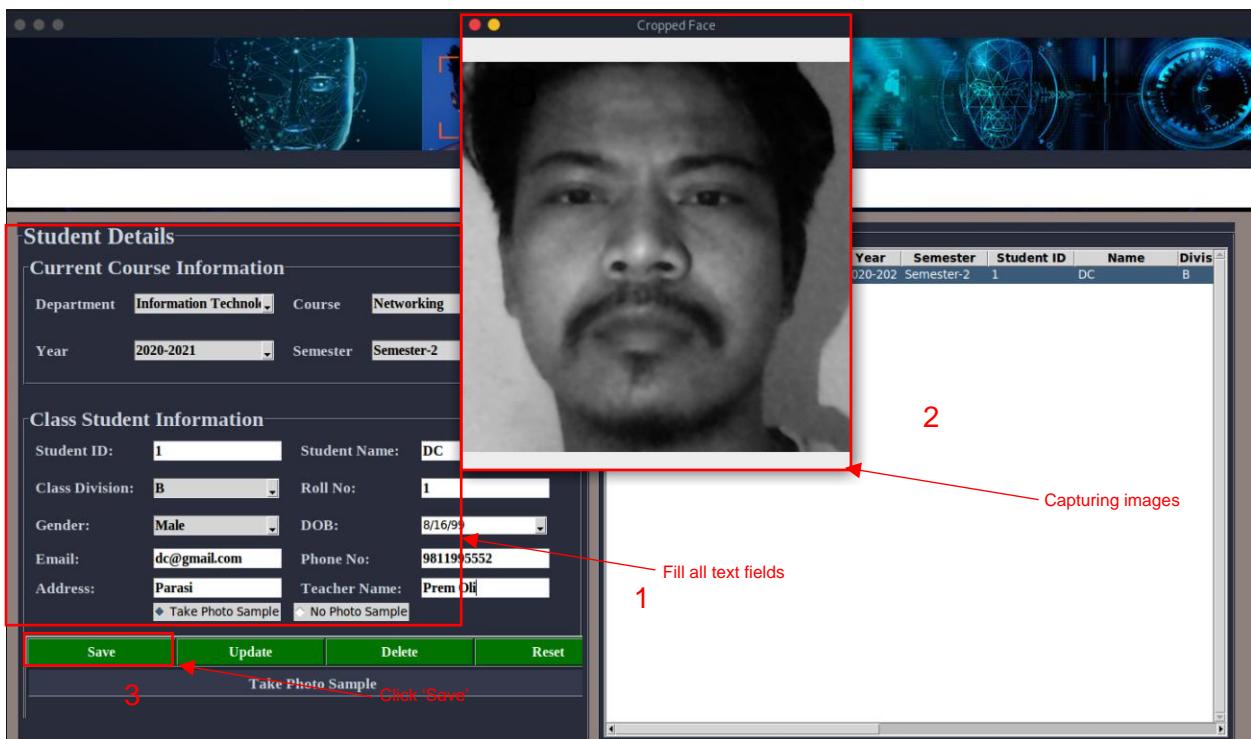


Figure 73: Test 06 (i)

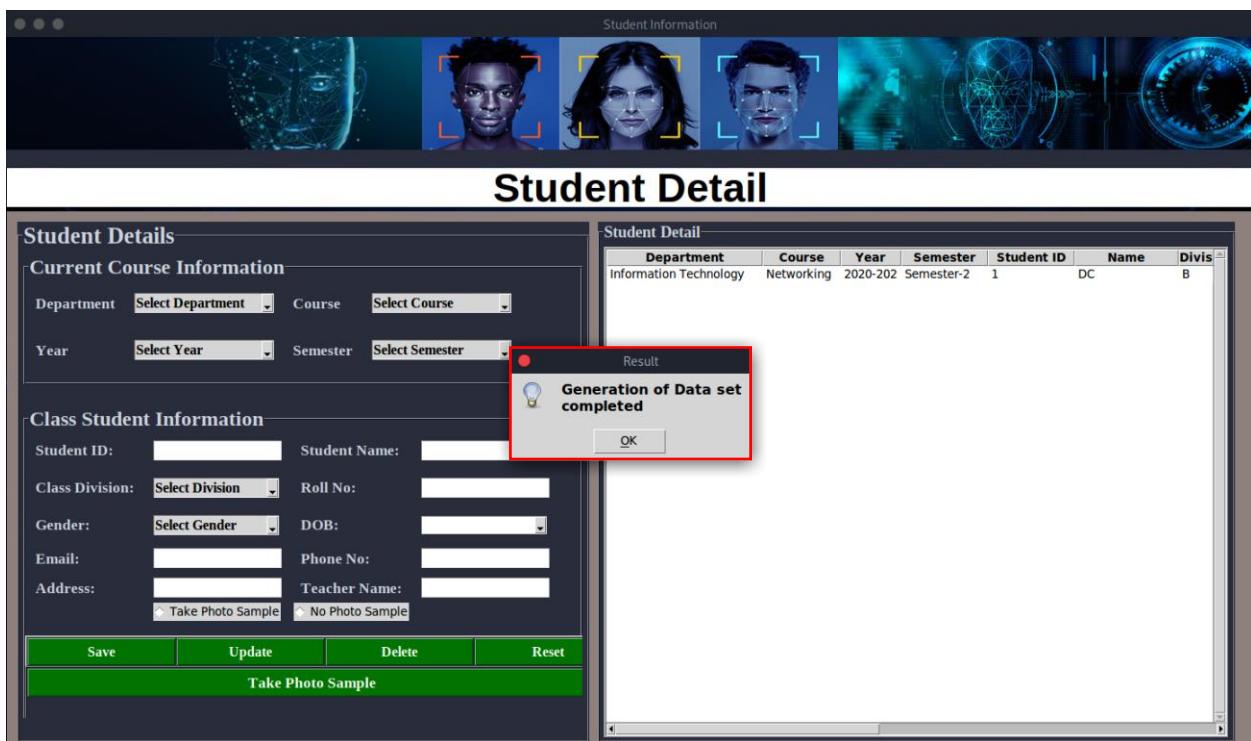


Figure 74: test 06 (ii)

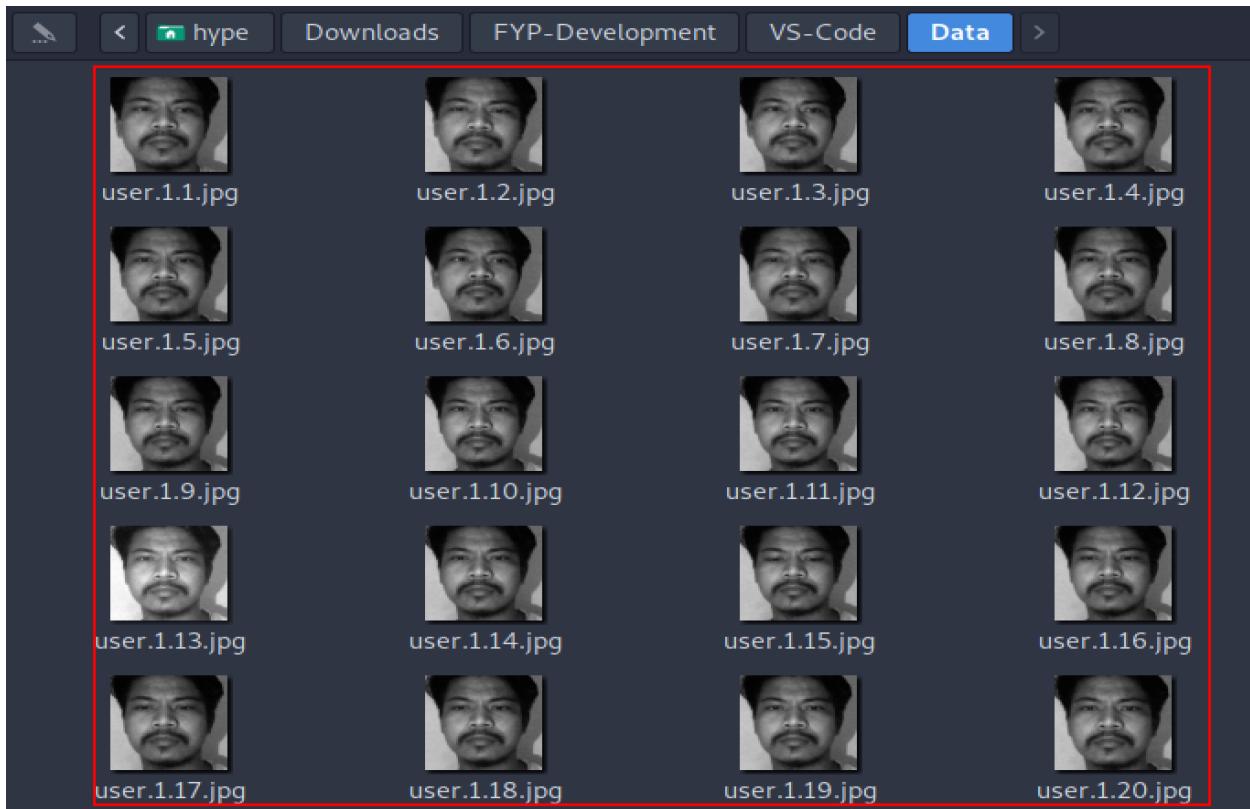


Figure 75: Test 06 (iii)

Captured images

The screenshot shows the phpMyAdmin interface for the "student" table in the "face\_recognizer" database. The table has columns: Department, Course, Year, Semester, StudentID, Name, Division, Roll No, Gender, DOB, Email, Phone, and Address. One row is selected for editing, with a red box around the "Edit" button. The row details are:

Department	Course	Year	Semester	StudentID	Name	Division	Roll No	Gender	DOB	Email	Phone	Address
Information Technology	Networking	2020-2021	Semester-2	1	DC	B	1	Male	8/16/99	dc@gmail.com	9811995552	Par

Figure 76: Test 06 (iv)

#### 4.2.1.7 Test 07: To update information of existing student

Test	07
Objective	To update information of existing student
Action	Update Contact Number of existing student
Expected Result	Student detail would be updated.
Actual Result	Student detail is updated.
Conclusion	Test successful.

Table 10: Test 07: To update information of existing student

The screenshot shows two windows side-by-side. The left window is titled 'Student Detail' and contains a form for 'Student Details'. It includes sections for 'Current Course Information' (Department: Information Technol., Course: Networking, Year: 2020-2021, Semester: Semester-2) and 'Class Student Information' (Student ID: 1, Student Name: DC, Class Division: B, Roll No: 1, Gender: Male, DOB: 8/16/99, Email: dc@gmail.com, Phone No: 9811995552, Address: Parasi, Teacher Name: Prem Oli). There are checkboxes for 'Take Photo Sample' and 'No Photo Sample'. The right window is also titled 'Student Detail' and displays a table with one row of data: Email: dc@gmail.com, Phone: 9811995552, Address: Parasi, Teacher: Prem Oli, and PhotoSampleStatus: Yes. A red box labeled '2' highlights the entire 'Student Details' form. A red box labeled '1' highlights the entire table in the right window. Red arrows point from the 'Update' button in the left form to the table in the right window, and another arrow points from the 'Click on student row' text to the table.

Email	Phone	Address	Teacher	PhotoSampleStatus
dc@gmail.com	9811995552	Parasi	Prem Oli	Yes

Figure 77: Test 07 (i)

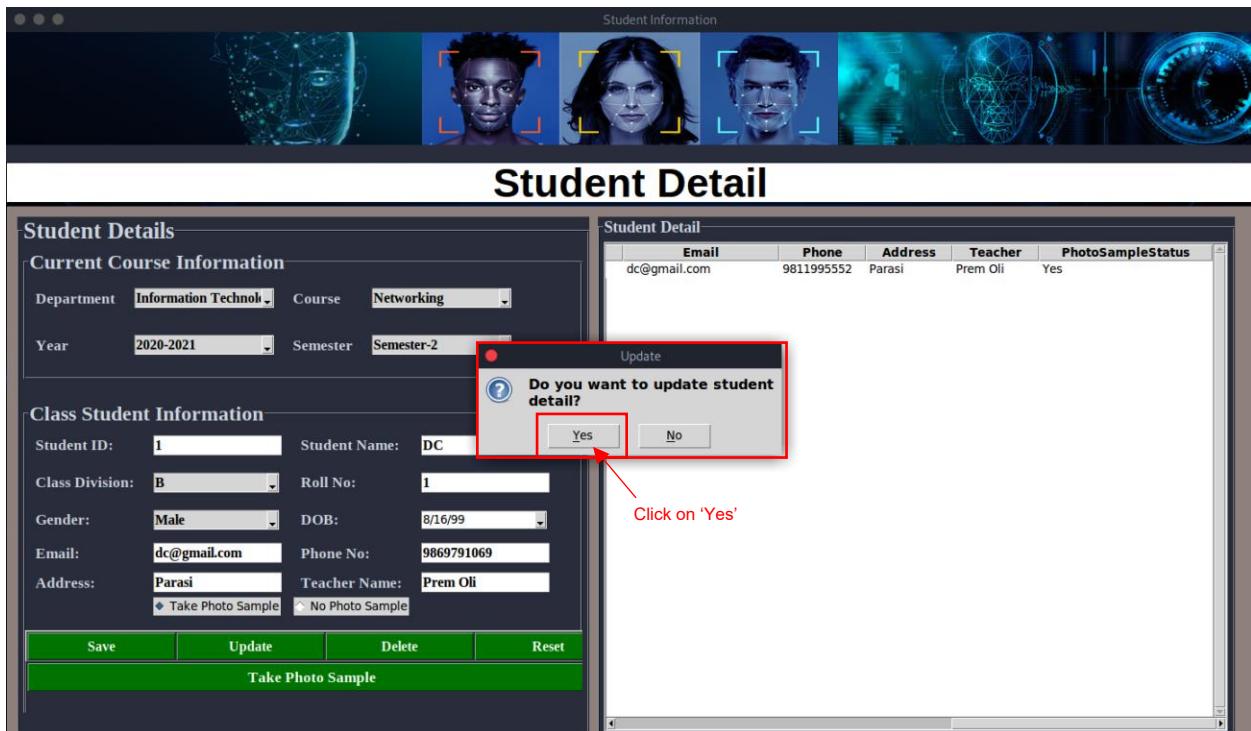


Figure 78: Test 07 (ii)

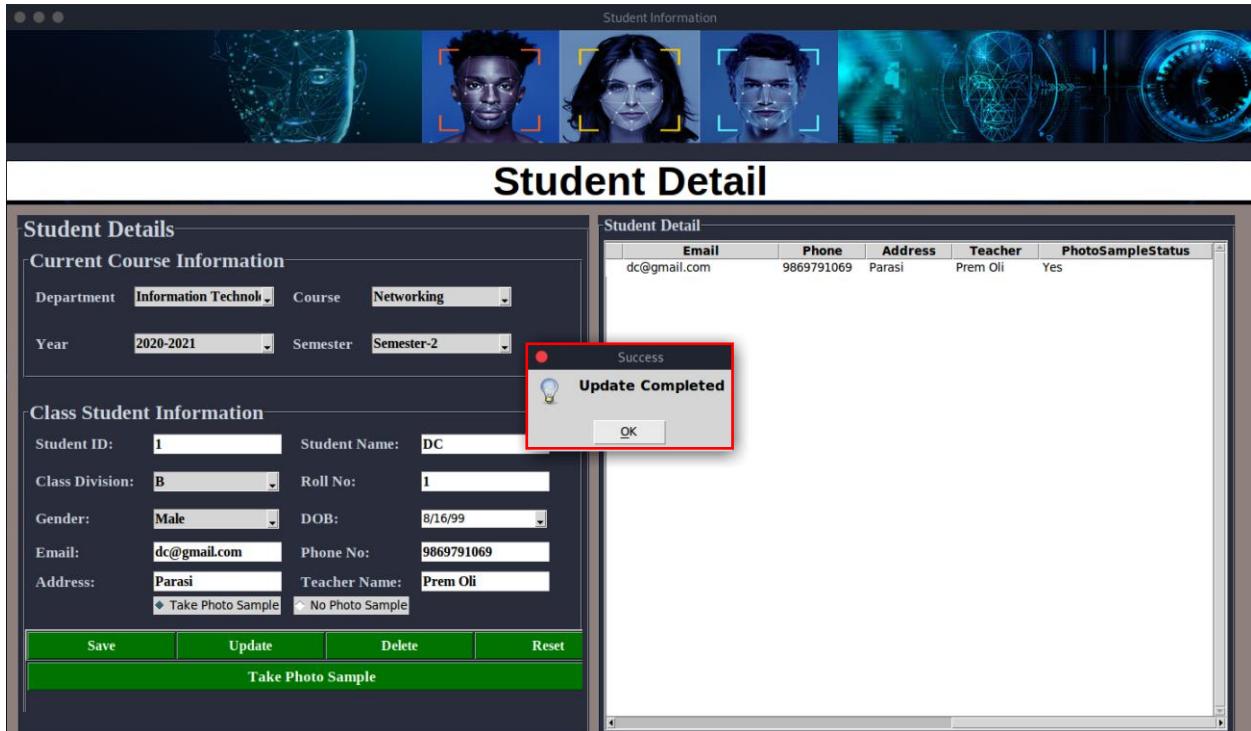


Figure 79: Test 07 (iii)

#### 4.2.1.8 Test 08: To reset filled-up information in text areas

Test	08
Objective	To reset filled-up information in text areas
Action	All the text area's information's are filled up and the clicked on 'Reset' button
Expected Result	All text field and combo box would be reset
Actual Result	All text filed and combo box is reset.
Conclusion	Test successful.

Table 11: Test 08: To reset filled-up information in text areas

**Student Details**

**Current Course Information**

Department	Information Technolo	Course	Networking
Year	2020-2021	Semester	Semester-2

**Class Student Information**

Student ID:	1	Student Name:	DC
Class Division:	B	Roll No:	1
Gender:	Male	DOB:	8/16/99
Email:	dc@gmail.com	Phone No:	9869791069
Address:	Parasi	Teacher Name:	Prem Oli
<input checked="" type="radio"/> Take Photo Sample <input type="radio"/> No Photo Sample			

Save    Update    Delete    **Reset**

**Take Photo Sample**

Click on 'Reset'

Figure 80: Test 08 (i)

**Student Details**

**Current Course Information**

Department  Course

Year  Semester

**Class Student Information**

Student ID:	<input type="text"/>	Student Name:	<input type="text"/>
Class Division:	<input type="button" value="Select Division"/>	Roll No.:	<input type="text"/>
Gender:	<input type="button" value="Select Gender"/>	DOB:	<input type="text"/>
Email:	<input type="text"/>	Phone No.:	<input type="text"/>
Address:	<input type="text"/>	Teacher Name:	<input type="text"/>
<input type="radio"/> Take Photo Sample		<input type="radio"/> No Photo Sample	

**Save      Update      Delete      Reset**

**Take Photo Sample**

Figure 81: Test 08 (ii)

#### 4.2.1.9 Test 09: To delete the existing student details

Test	09
Objective	To delete the existing student details
Action	Clicked on specific student in right frame and followed with click on 'Delete' button
Expected Result	Student details is expected to be deleted.
Actual Result	Student details is deleted successfully.
Conclusion	Test successful.

Table 12: Test 09: To delete the existing student details

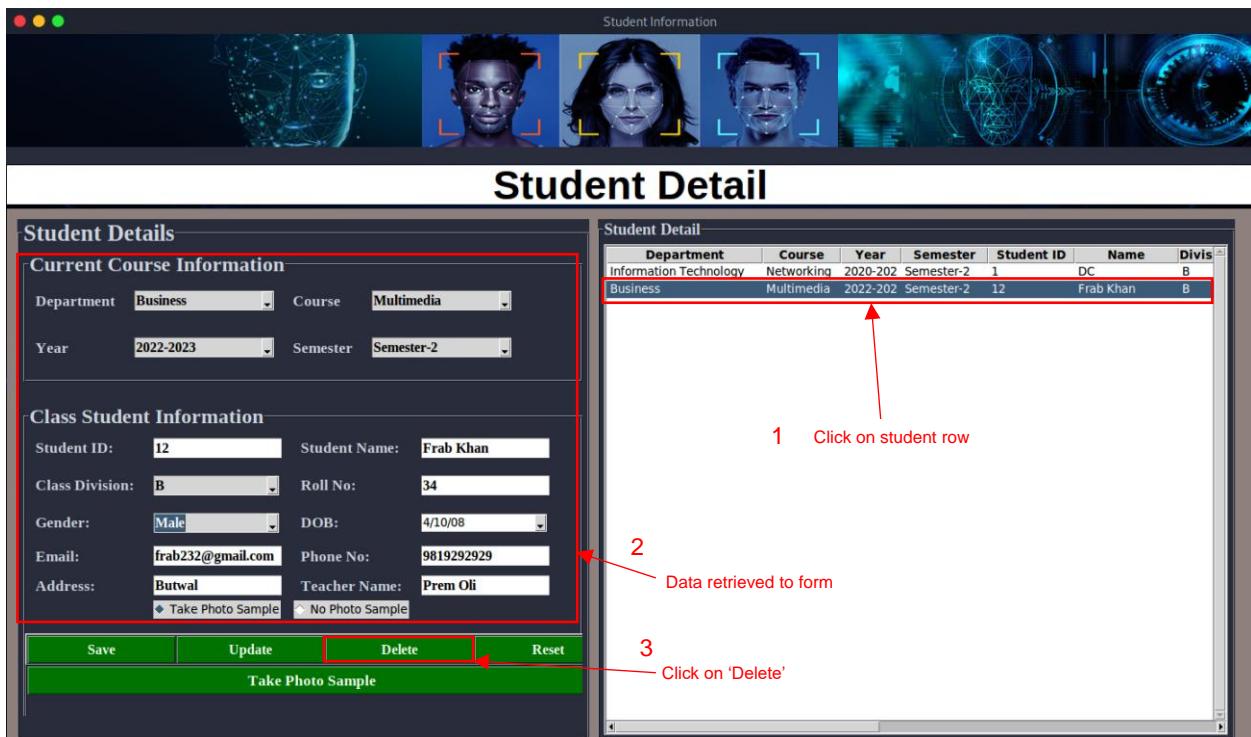


Figure 82: Test 09 (i)

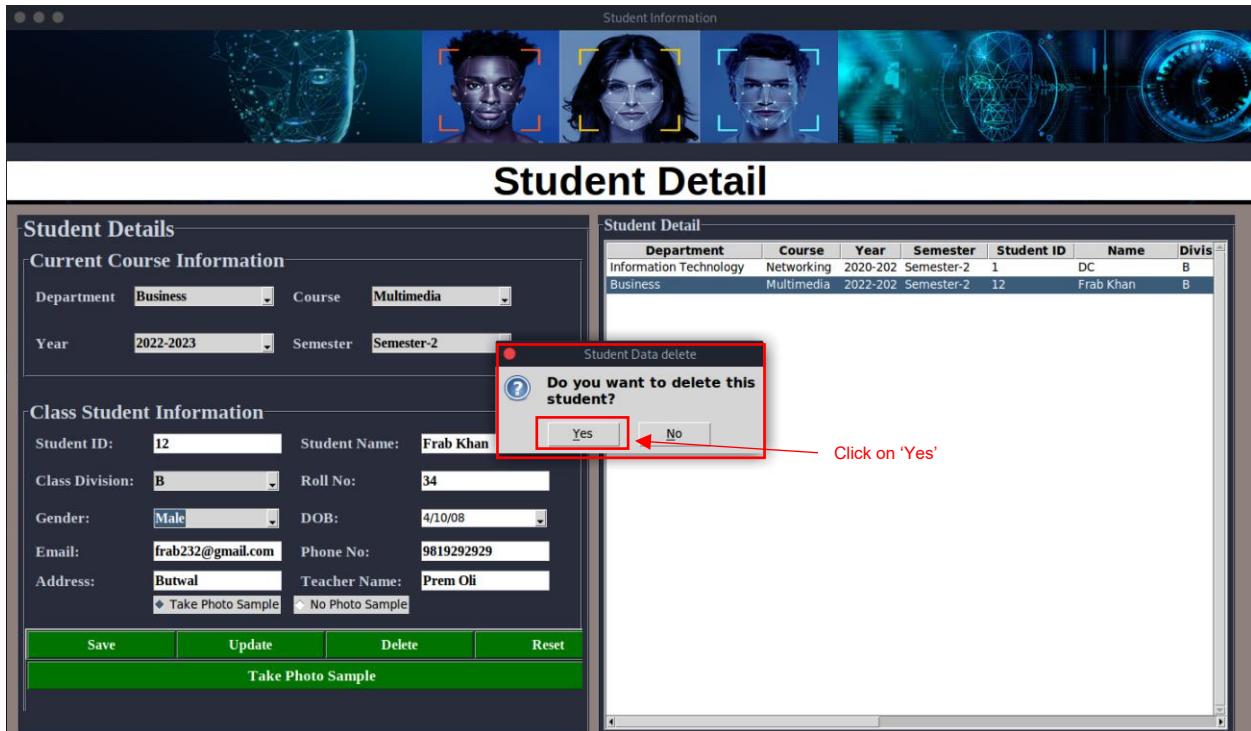


Figure 83: Test 09 (ii)

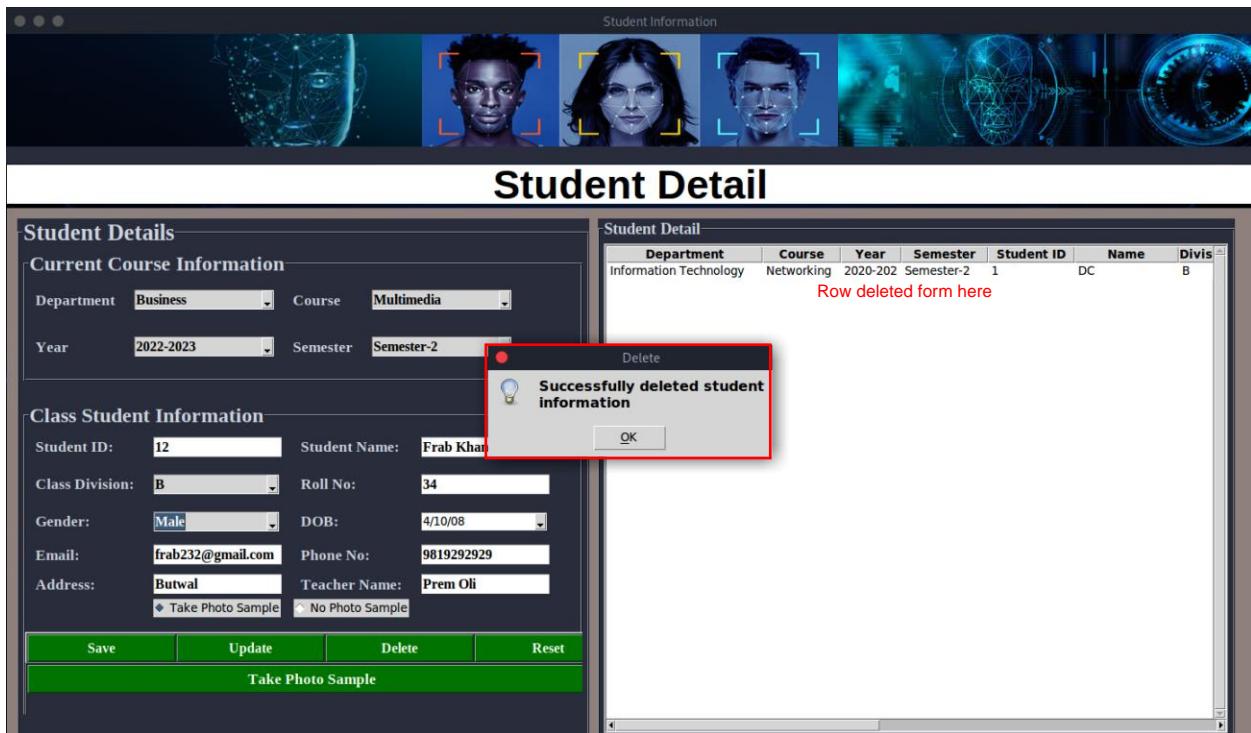


Figure 84: Test 09 (iii)

#### 4.2.1.10 Test 10: To Train all dataset in database

Test	10
Objective	To Train all dataset in database
Action	In Train window, clicked on 'Train' button
Expected Result	All datasets are expected to trained successfully.
Actual Result	All dataset is trained successfully.
Conclusion	Test successful.

Table 13: Test 10: To Train all dataset in database

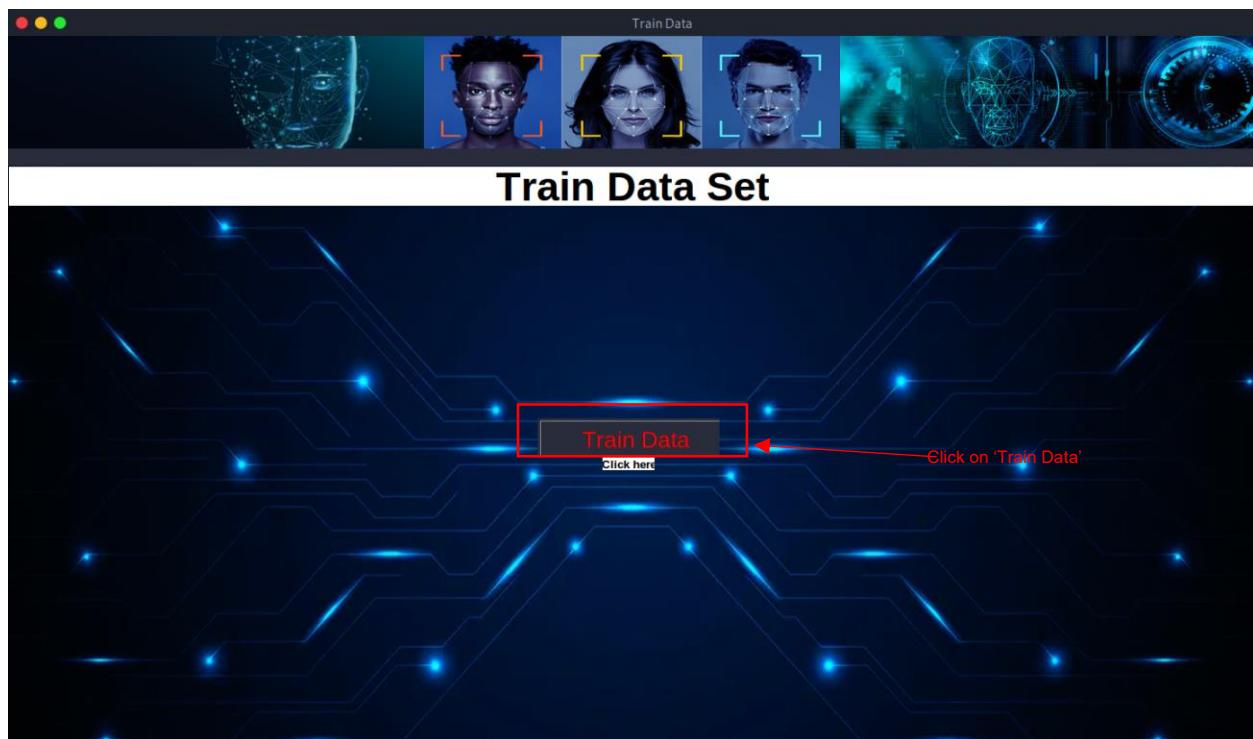


Figure 85: Test 10 (i)

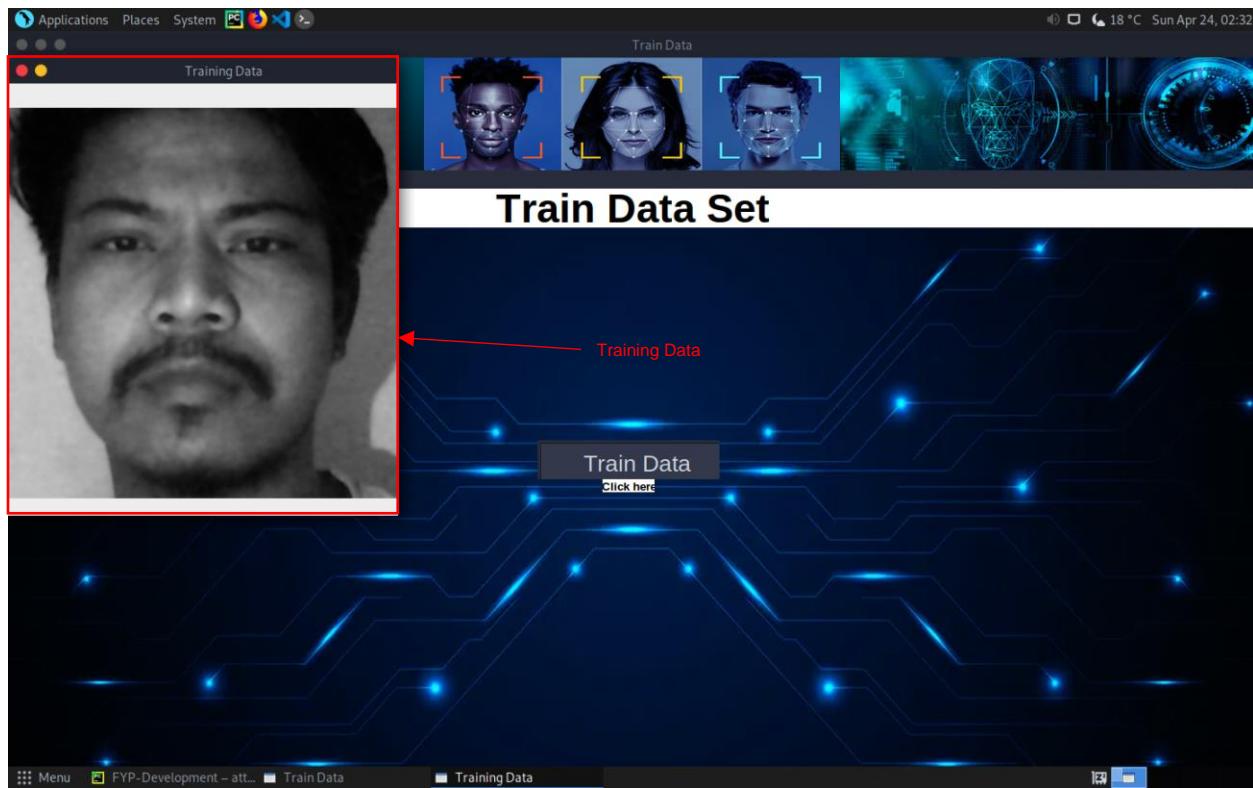


Figure 86: Test 10 (ii)

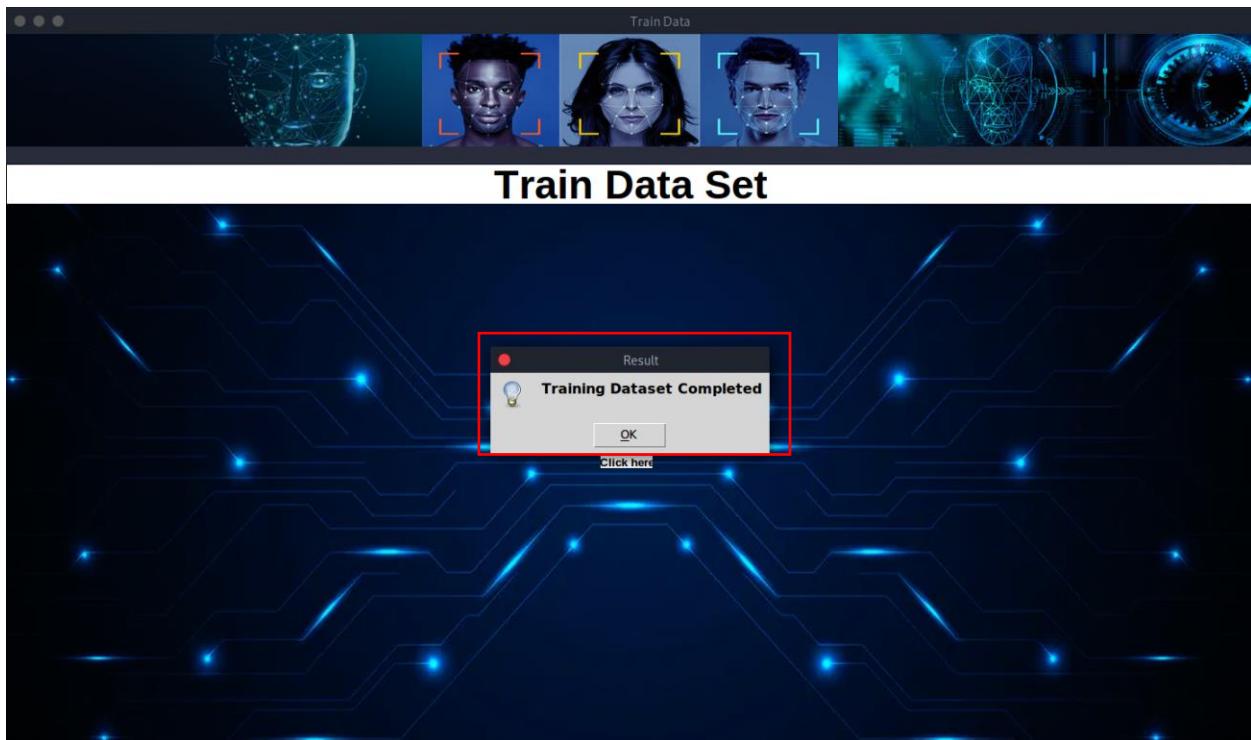


Figure 87: Test 10 (iii)

#### 4.2.1.11 Test 11: To start face recognition from ‘Face Recognition’ window

Test	11
Objective	To start face recognition from ‘Face Recognition’ window
Action	In ‘Face Recognition’ windows, click on ‘Face Recognize’ button.
Expected Result	A camera is expected to launch and start recognizing face with displaying ID and Name on top of face.
Actual Result	A camera is launched and started recognizing faces with displaying ID and Name on top of face.
Conclusion	Test successful.

Table 14: Test 11: To start face recognition from ‘Face Recognition’ window

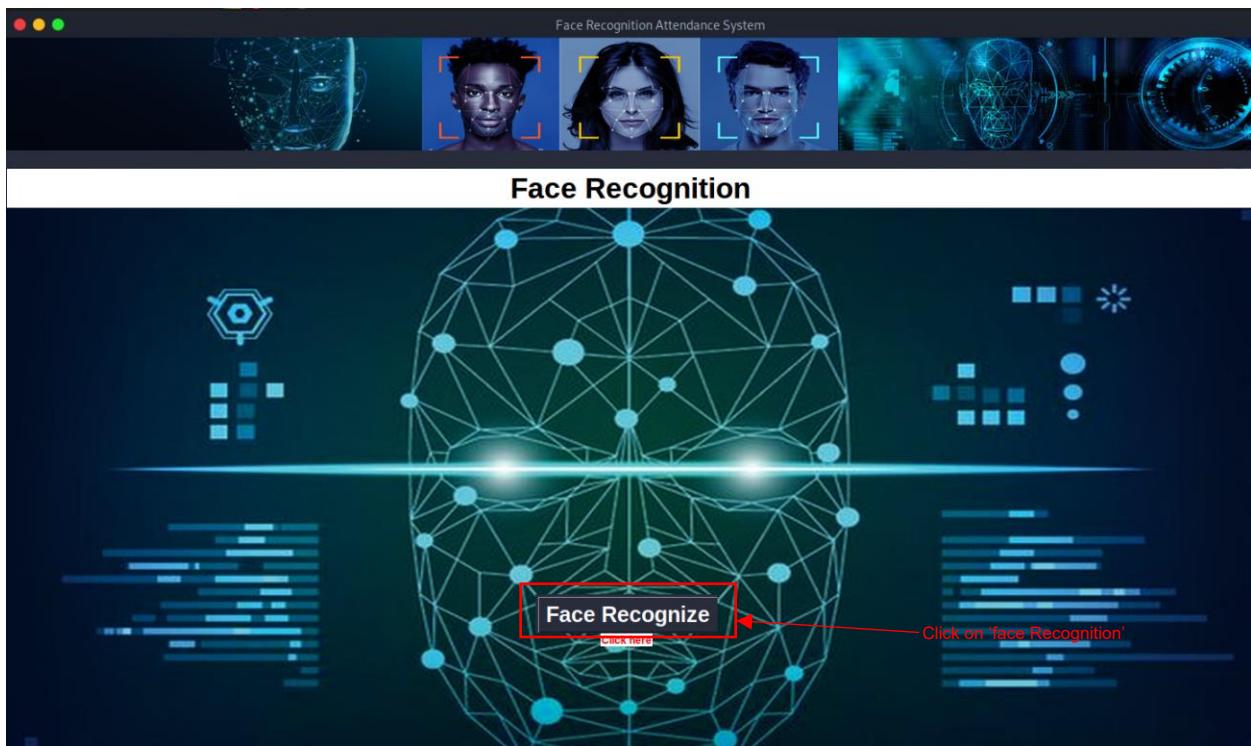


Figure 88: Test 11 (i)

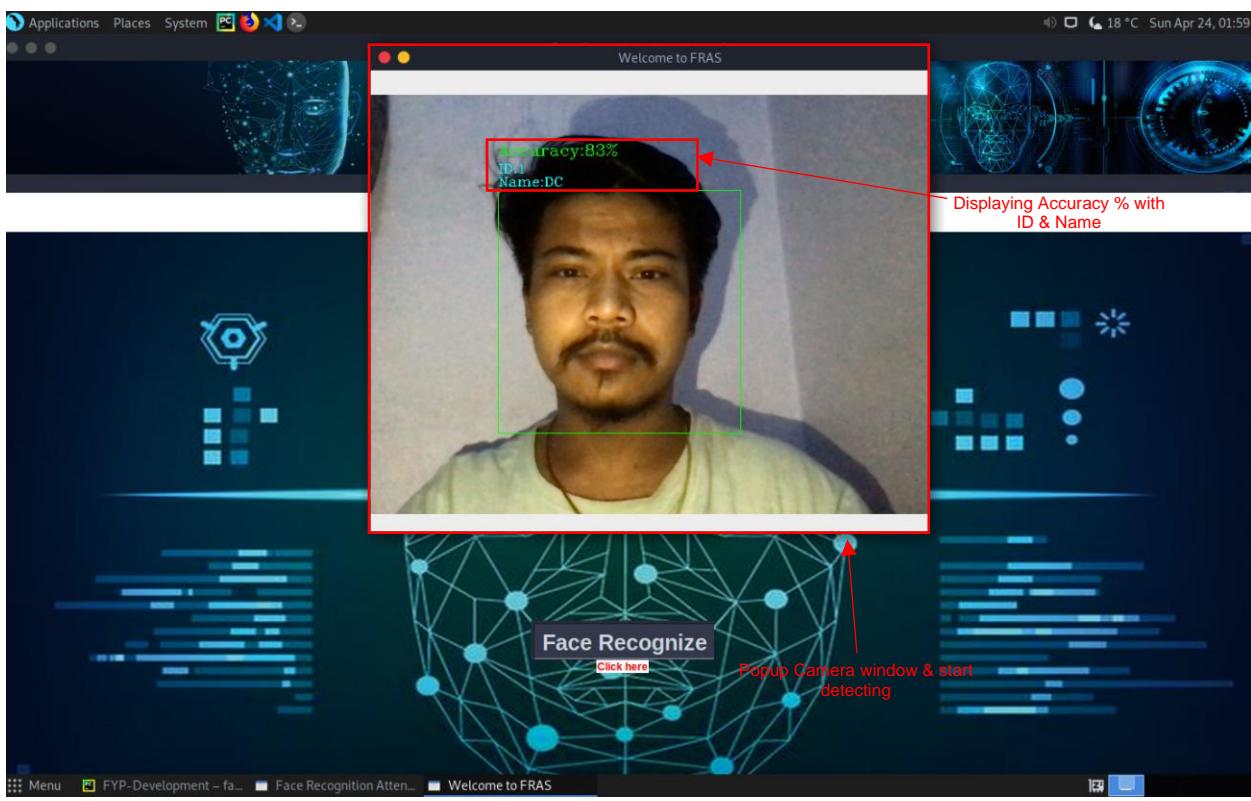


Figure 89: Test 11 (ii)

#### 4.2.1.12 Test 12: To verify attendance marked in csv file

Test	12
Objective	To verify attendance marked in csv file
Action	After face recognition, attendance is automatically marked into csv file
Expected Result	Attendance is supposed to be marked present in csv file
Actual Result	Attendance is marked present in csv file
Conclusion	Test successful

Table 15: Test 12: To verify attendance marked in csv file

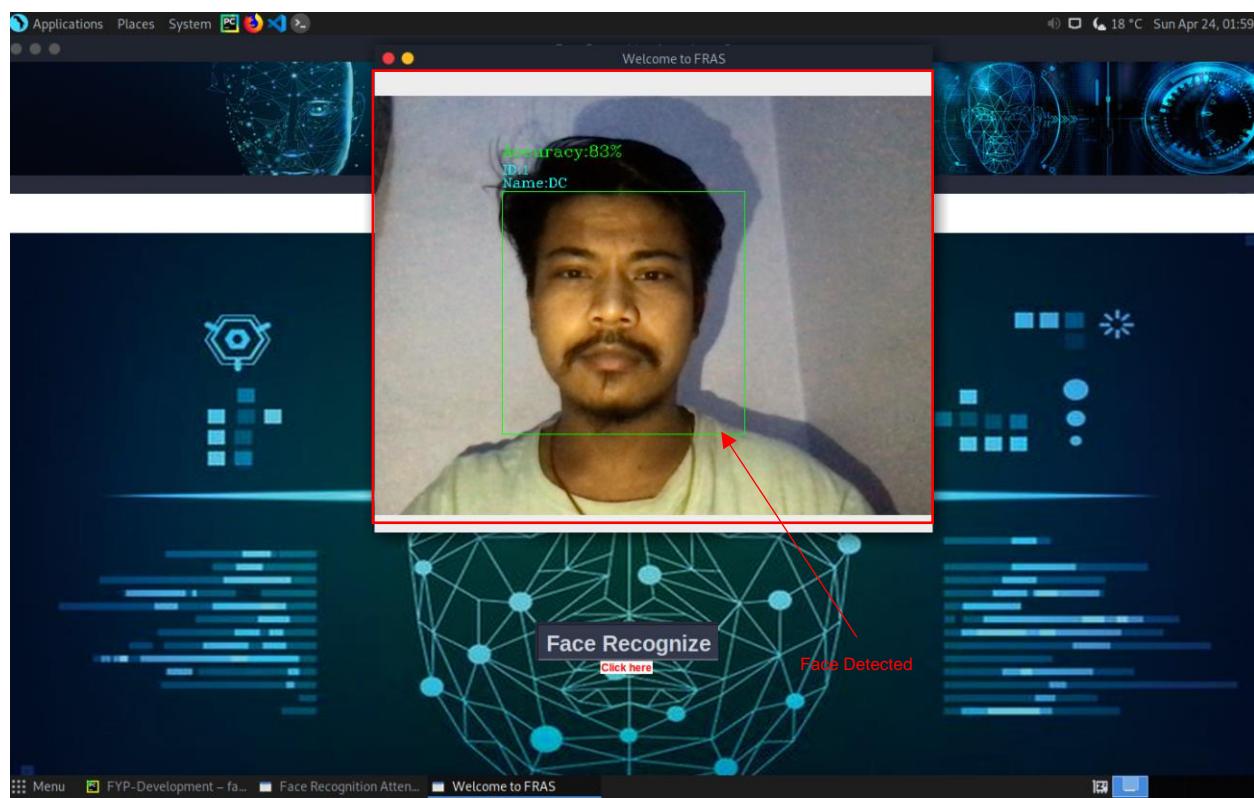


Figure 90: Test 12 (i)

```

1
2 1,1,DC,Information Technology,14:28:55,12/04/2022,Present
3
4
5
6

```

Figure 91: Test 12 (ii)

#### 4.2.1.13 Test 13: To import csv file into ‘attendance’ window

Test	12
Objective	To import csv file into ‘attendance’ window
Action	Clicked on ‘Import CSV’ button and selected Attendance CSV file
Expected Result	It is supposed to import CSV file to this window.
Actual Result	It successfully imported CSV file to this window.
Conclusion	Test successful.

Table 16: Test 13: To import csv file into ‘attendance’ window

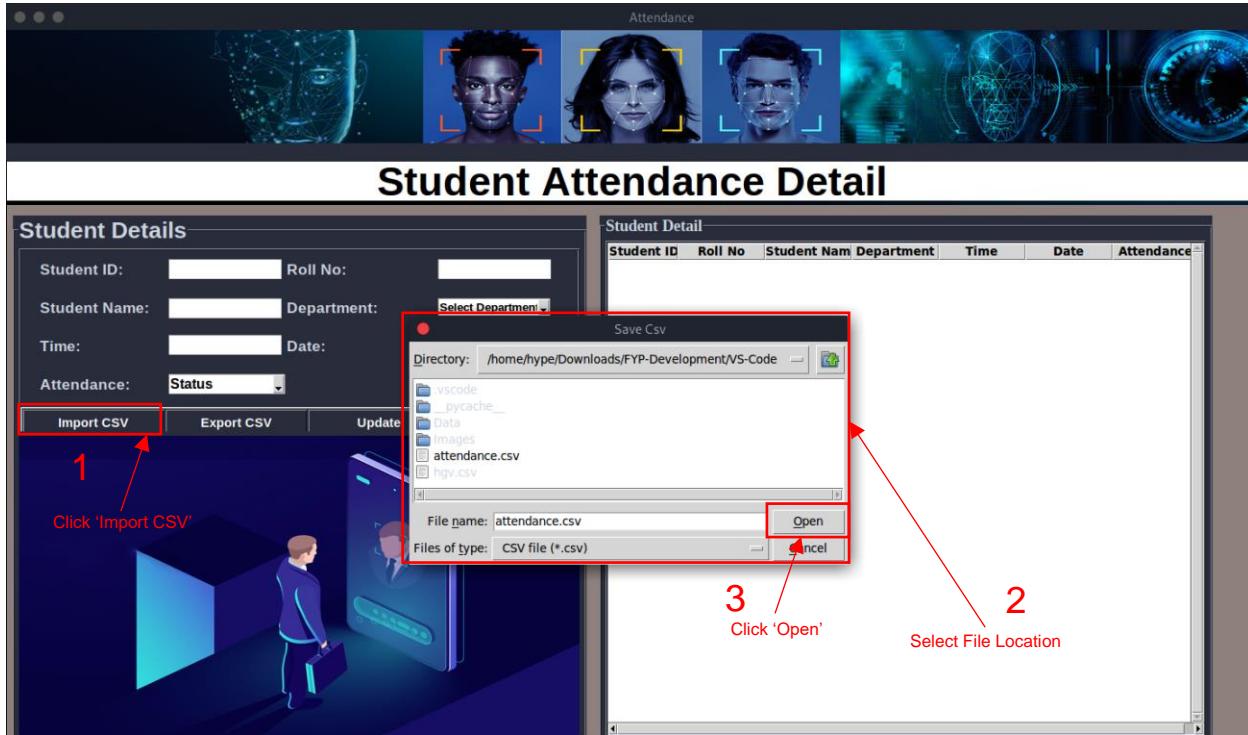


Figure 92: Test 13 (i)

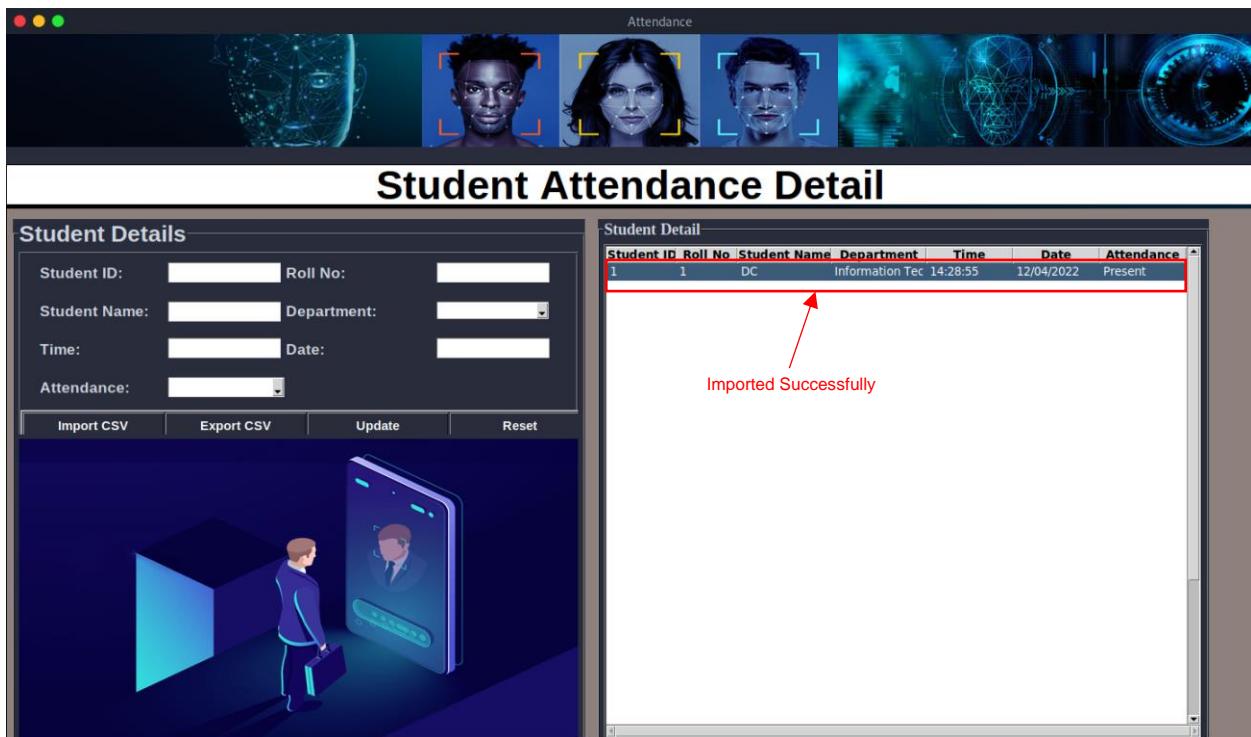


Figure 93: Test 13 (ii)

#### 4.2.1.14 Test 14: To modify CSV attendance file and export

Test	13
Objective	To modify CSV attendance file and export
Action	Necessary modification followed by export CSV file on clicking 'Export CSV' button
Expected Result	Attendance is supposed to be modified and export to specified path and folder.
Actual Result	Attendance is modified and exported to specified path and folder.
Conclusion	Test successful.

Table 17: Test 14: To modify CSV attendance file and export

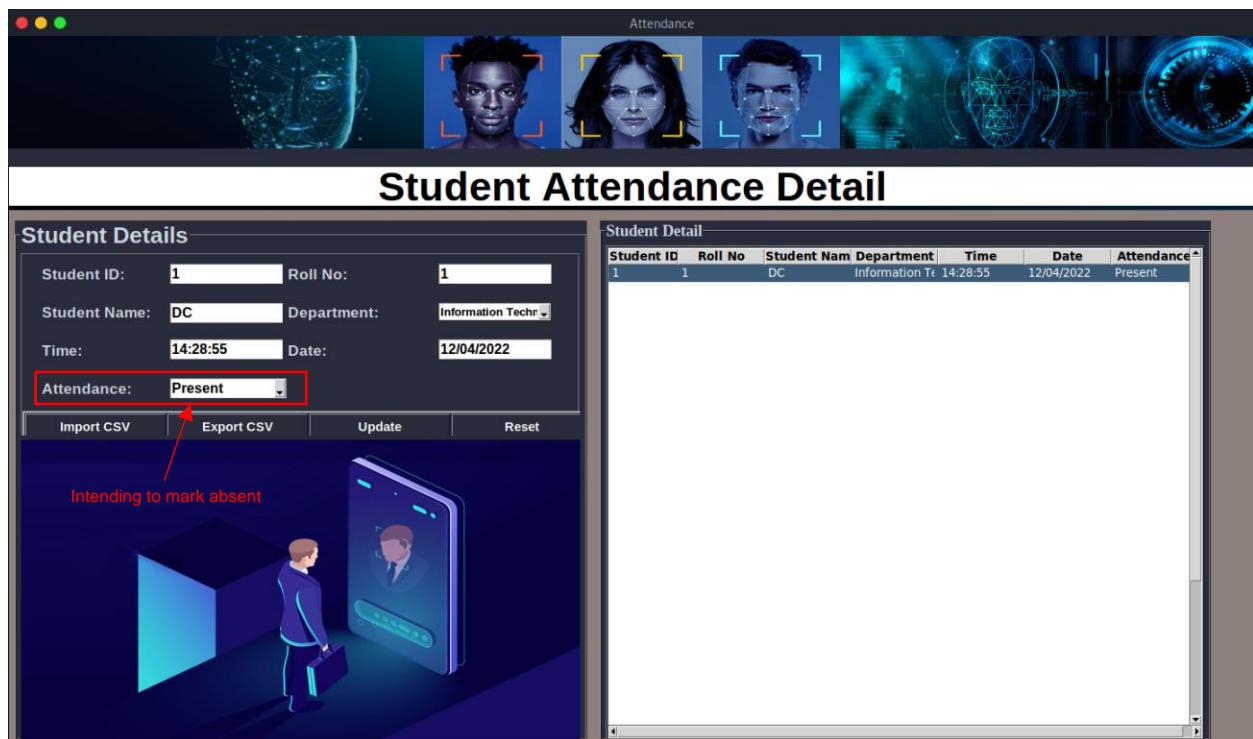


Figure 94: Test 14 (i)

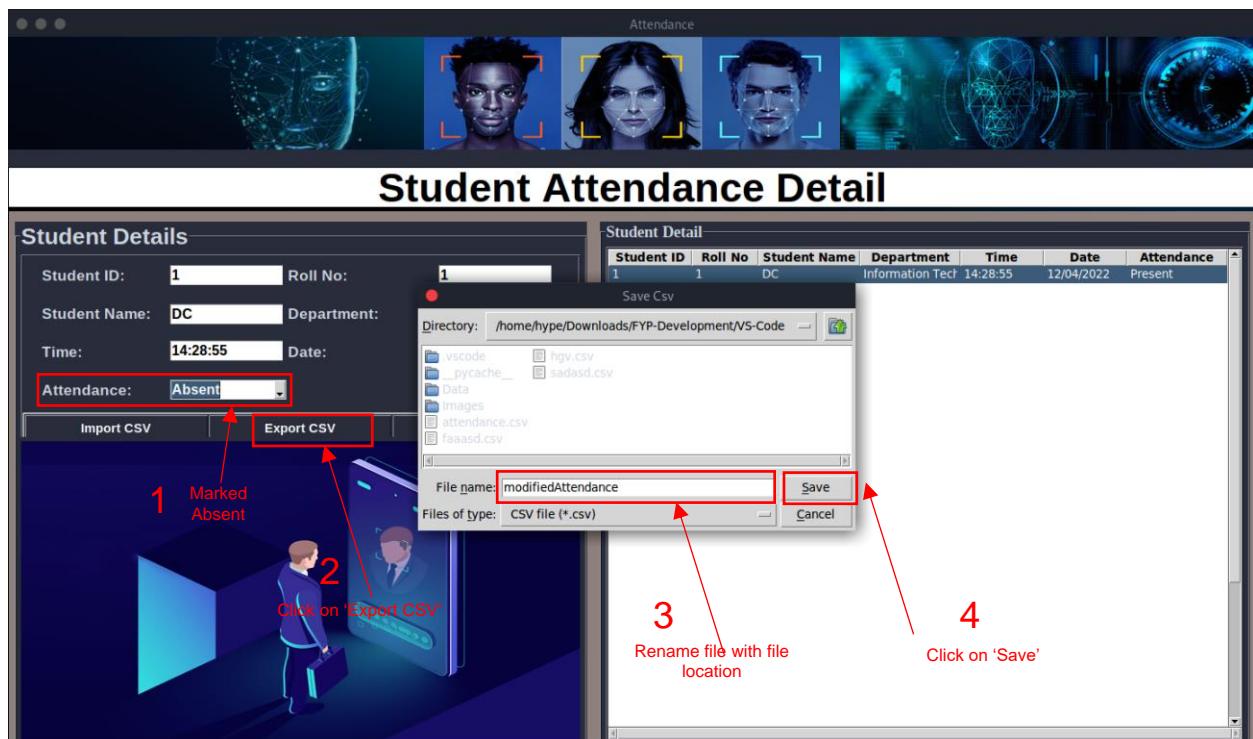


Figure 95: Test 14 (ii)

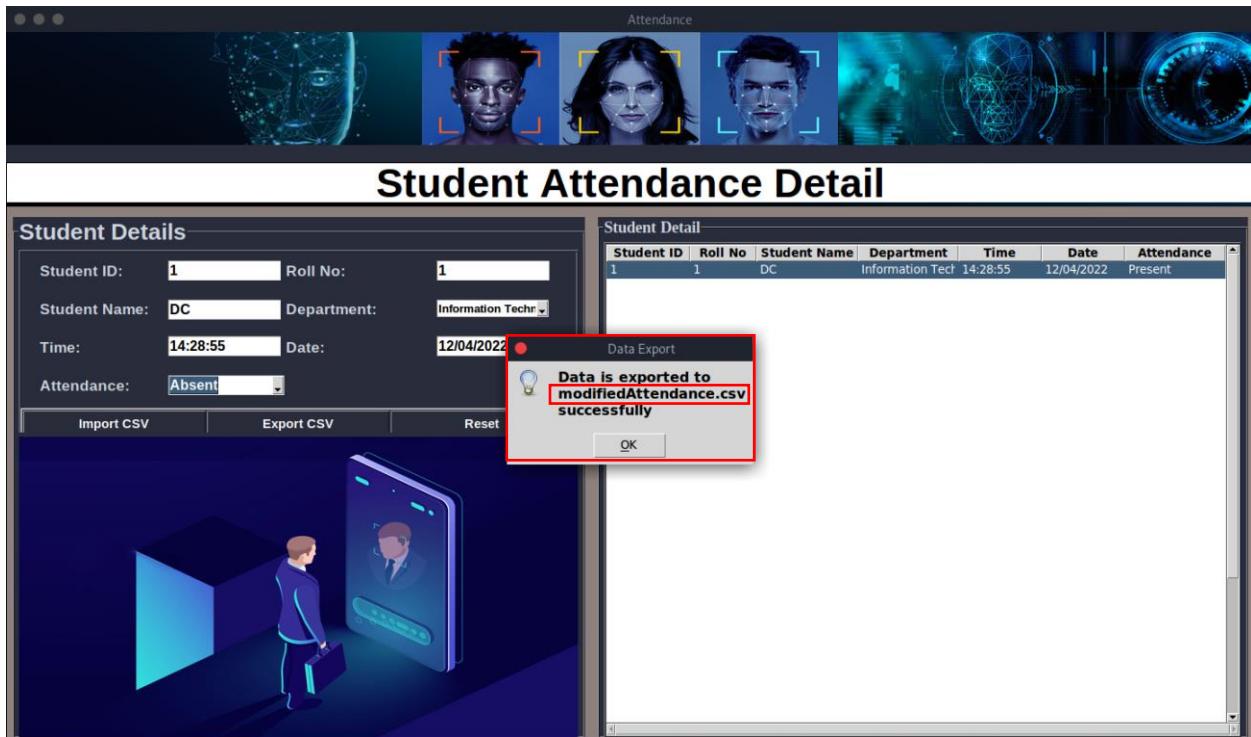


Figure 96: Test 14 (iii)

#### 4.2.1.15 Test 15: To reset all text field on single click

Test	14
Objective	To reset all text field on single click
Action	Click on 'Reset'
Expected Result	All fields are supposed to be reset to blank.
Actual Result	All fields are reset to blank successfully.
Conclusion	Test successful.

Table 18: Test 15: To reset all text field on single click

### Student Details

Student ID:	1	Roll No:	1
Student Name:	DC	Department:	Information Techn
Time:	14:28:55	Date:	12/04/2022
Attendance:	Absent		

**Import CSV**   **Export CSV**   **Reset**

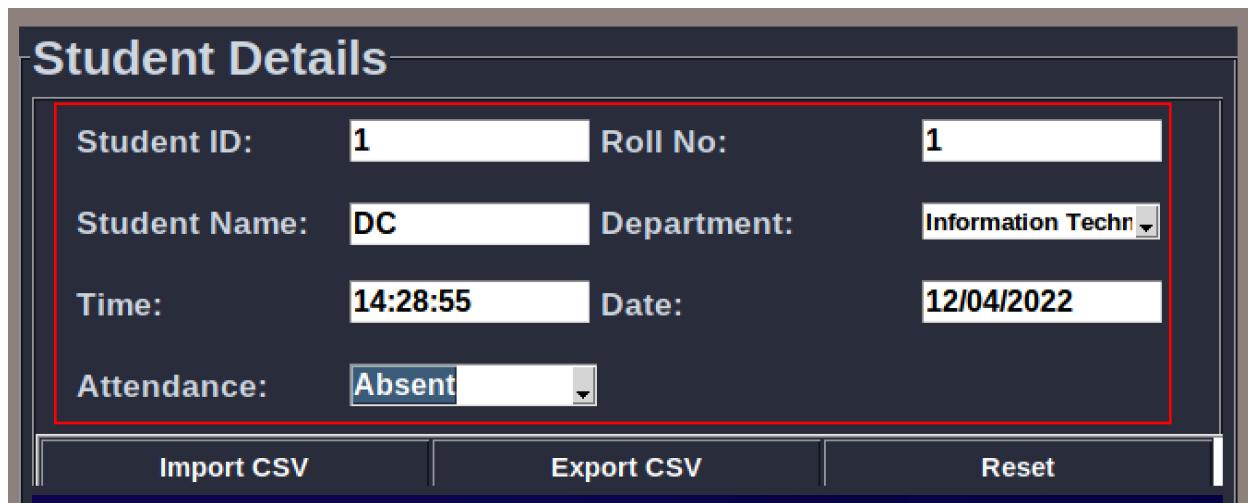


Figure 97: Test 15 (i)

### Student Details

Student ID:		Roll No:	
Student Name:		Department:	
Time:		Date:	
Attendance:			

**Import CSV**   **Export CSV**   **Reset**

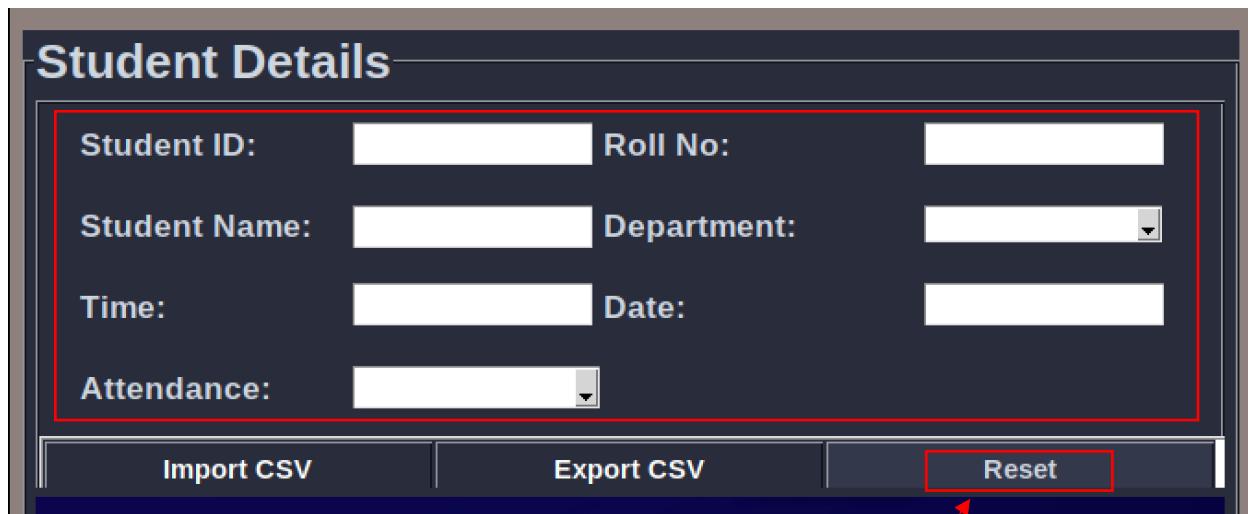


Figure 98: Test 15 (ii)

All fields reset on  
single click



### 4.3 Black Box Testing/Failure Test Cases

#### 4.3.1 Error 01- In teacher registration, strong password is not recommended.

During the process of registration, strong password was not recommended and system can accept simple password of numbers or text only.

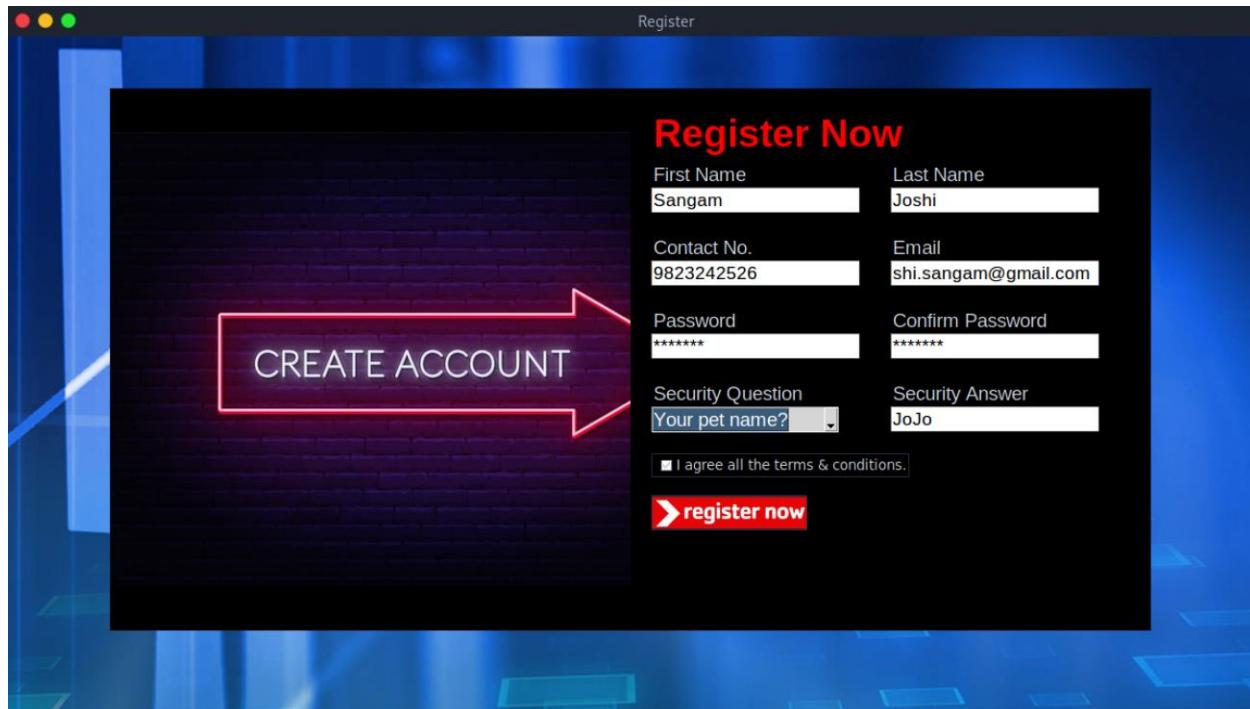


Figure 99: Error 01 (i)

	First Name	Last Name	Contact No.	Email	Password	ConfirmPassword	SecurityQuestion	SecurityAns
<input checked="" type="checkbox"/> Edit	Sangam	Joshi	982324256	joshi.sangam@gmail.com	222222	222222	Your pet name?	Jojo
<input type="checkbox"/> Edit	Prem	Oli	9812223242	prem.olii52@gmail.com	Prem@53	Prem@52	Your pet name?	Sniffy

Figure 100: Error 01 (ii)

**Error in code:** There is no any code of password validation.

```

#Function==Declaration
def register_data(self):
    if self.var_first_name.get()=="" or self.var_email.get()=="" or self.var_security_qst.get()=="Select" or self.var_contact.get()=="":
        messagebox.showerror("Error","All fields are required")
    elif self.var_pass.get()!= self.var_confirm_pass.get():
        messagebox.showerror("Error","Unmatched password")
    elif self.var_check_button.get()==0:
        messagebox.showerror("Error","Must agree terms & conditions")
    elif not ("@" or ".com") in self.var_email.get():
        messagebox.showerror("Error occurred","Invalid email! Please Enter email like <santos@gmail.com>",parent=self.root)
    else:
        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
        my_cursor=conn.cursor()
        query="Select * from register where email=%s"
        value=(self.var_email.get(),)
        my_cursor.execute(query,value)
        fetched_data=my_cursor.fetchone()
        if fetched_data==None:
            messagebox.showerror("Error", "Existing User, Try another email")
        else:
            my_cursor.execute("INSERT INTO register VALUE(%s,%s,%s,%s,%s,%s)",(
                self.var_first_name.get(),
                self.var_last_name.get(),
                self.var_contact.get(),
                self.var_email.get(),
                self.var_pass.get(),
                self.var_confirm_pass.get(),
                self.var_security_qst.get(),
                self.var_security_ans.get()
            ))

```

Figure 101: Error 01 (iii)

**Solution to the code:** Password validation is set up.

```
#Function==Declaration
def register_data(self):
    if self.var_first_name.get()=="" or self.var_email.get()=="" or self.var_security_qst.get()=="Select" or self.var_conta
        messagebox.showerror("Error","All fields are required")
    elif self.var_pass.get()!= self.var_confirm_pass.get():
        messagebox.showerror("Error","Unmatched password")
    elif self.var_check_button.get()==0:
        messagebox.showerror("Error","Must agree terms & conditions")
    elif not ("@" or ".com") in self.var_email.get():
        messagebox.showerror("Error occurred","Invalid email! Please Enter email like <santos@gmail.com>",parent=self.root)
    elif not ("@" or "!" or "$" or "-" or "." or "#") in self.var_pass.get():
        messagebox.showerror("Password Error","Please, Enter strong password like SanTos@123",parent=self.root)
    else:
        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
        my_cursor=conn.cursor()
        query="Select * from register where email=%s"
        value=(self.var_email.get(),)
        my_cursor.execute(query,value)
        fetched_data=my_cursor.fetchone()
        if fetched_data!=None:
            messagebox.showerror("Error","Existing User, Try another email")
        else:
```

Figure 102: Error 01 (iv)

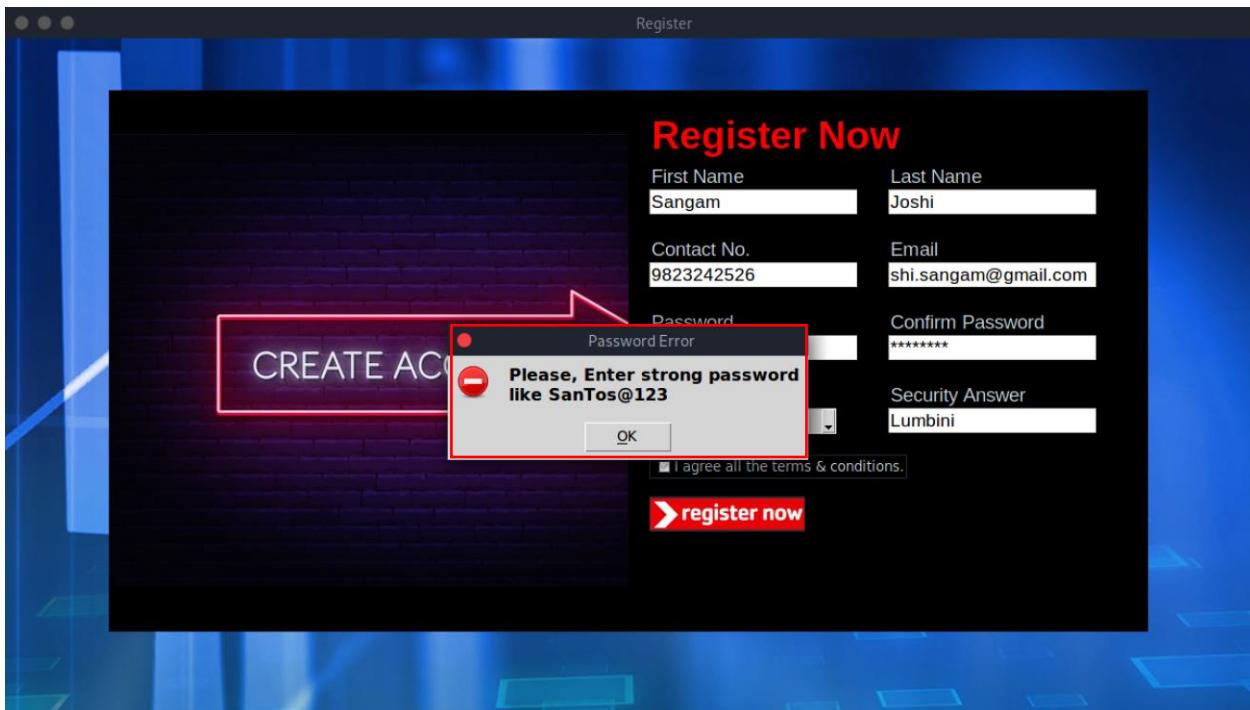


Figure 103: Error 01 (v)

**Solution Justification:**

Password validation was lacking in the system, where solution is the addition of password validation code.

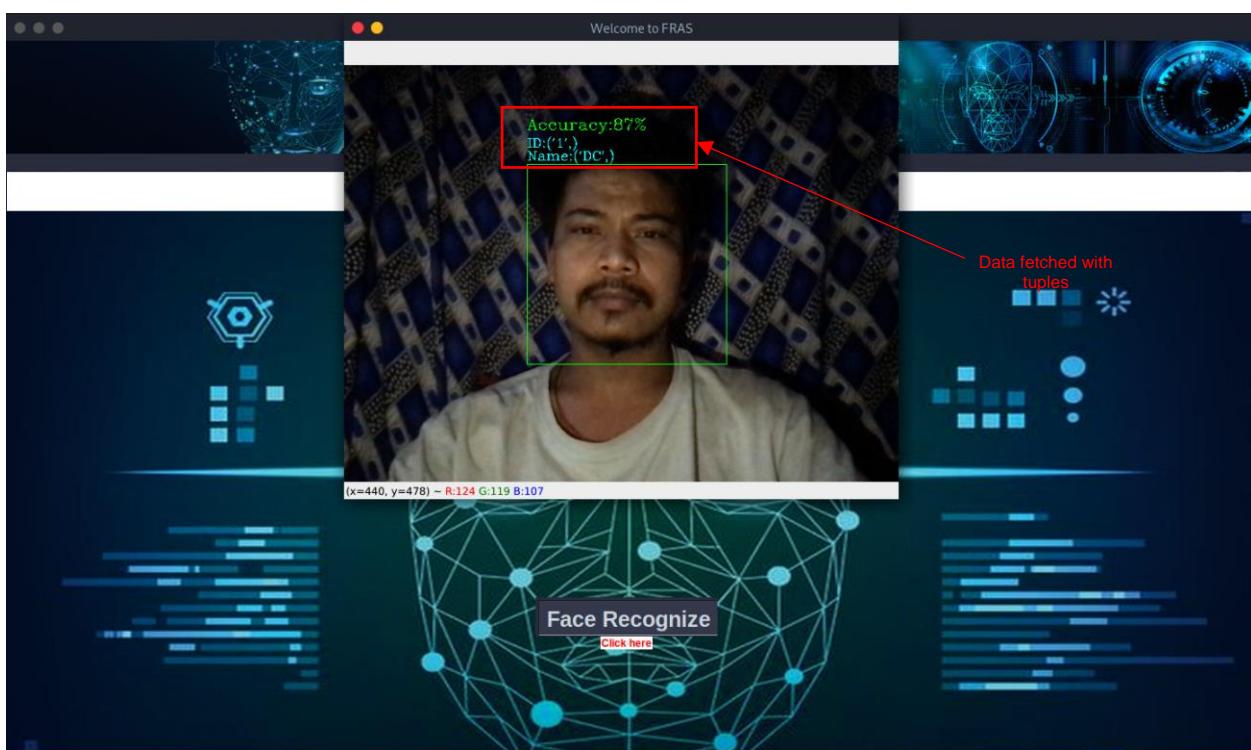
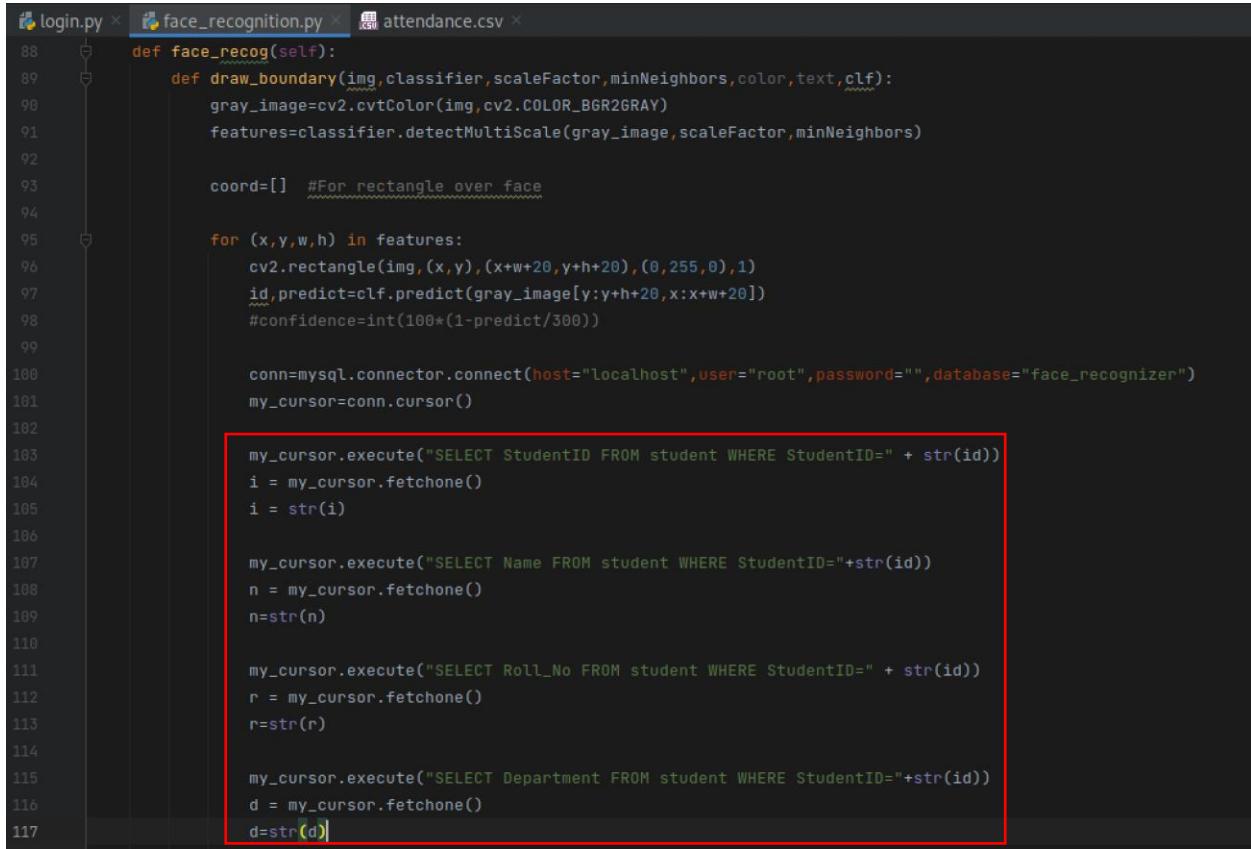
**4.3.2 Error 02- Data fetched from database while face recognition is included with Tuples**

Figure 104: Error 02 (i)

The screenshot shows a terminal window with three tabs: "login.py", "face\_recognition.py", and "attendance.csv". The "attendance.csv" tab is active, displaying a single row of data: "1", "('1',), (1,), ('DC',), ('Information Technology',), 22:35:38, 24/04/2022, Present". A red callout box highlights this row. A red arrow points from the text "Attendance marked with tuples" at the bottom right towards the highlighted row.

Figure 105: Error 02 (ii)

**Error in Code:** Fetchone attribute was fetching tuples from database.



```
login.py × face_recognition.py × attendance.csv ×
88 def face_recog(self):
89     def draw_boundary(img,classifier,scaleFactor,minNeighbors,color,text,clf):
90         gray_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
91         features=classifier.detectMultiScale(gray_image,scaleFactor,minNeighbors)
92
93         coord=[] #For rectangle over face
94
95         for (x,y,w,h) in features:
96             cv2.rectangle(img,(x,y),(x+w+20,y+h+20),(0,255,0),1)
97             id,predict=clf.predict(gray_image[y:y+h+20,x:x+w+20])
98             #confidence=int(100*(1-predict/300))
99
100            conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
101            my_cursor=conn.cursor()
102
103            my_cursor.execute("SELECT StudentID FROM student WHERE StudentID=" + str(id))
104            i = my_cursor.fetchone()
105            i = str(i)
106
107            my_cursor.execute("SELECT Name FROM student WHERE StudentID="+str(id))
108            n = my_cursor.fetchone()
109            n=str(n)
110
111            my_cursor.execute("SELECT Roll_No FROM student WHERE StudentID=" + str(id))
112            r = my_cursor.fetchone()
113            r=str(r)
114
115            my_cursor.execute("SELECT Department FROM student WHERE StudentID="+str(id))
116            d = my_cursor.fetchone()
117            d=str(d)
```

Figure 106: Error 02 (iii)

**Solution in code:** [index] in fetchone attribute excluded tuples while fetching data.

```

login.py x face_recognition.py x attendance.csv x
91         features=classifier.detectMultiScale(gray_image,scaleFactor,minNeighbors)
92
93         coord=[] #For rectangle over face
94
95     for (x,y,w,h) in features:
96         cv2.rectangle(img,(x,y),(x+w+20,y+h+20),(0,255,0),1)
97         id,predict=clf.predict(gray_image[y:y+h+20,x:x+w+20])
98         #confidence=int(100*(1-predict/300))
99
100        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
101        my_cursor=conn.cursor()
102
103        my_cursor.execute("SELECT StudentID FROM student WHERE StudentID=" + str(id))
104        i = my_cursor.fetchone()[0]
105        i = str(i)
106
107        my_cursor.execute("SELECT Name FROM student WHERE StudentID="+str(id))
108        n = my_cursor.fetchone()[0]
109        n=str(n)
110
111        my_cursor.execute("SELECT Roll_No FROM student WHERE StudentID=" + str(id))
112        r = my_cursor.fetchone()[0]
113        r=str(r)
114
115        my_cursor.execute("SELECT Department FROM student WHERE StudentID="+str(id))
116        d = my_cursor.fetchone()[0]
117        d=str(d)

```

Figure 107: Error o2 (iv)

### Solution Justification:

In python fetchone fetches the next row (case) from the active dataset. In this case, fetchone () was fetching dataset but with tuples too but the solution fetchone ()[0] fetches dataset excluding tuples.

#### 4.3.3 Error 03- Unable to launch camera

When clicking on ‘Face recognize’ button in ‘Face Recognition’ window, camera was unable to launch.

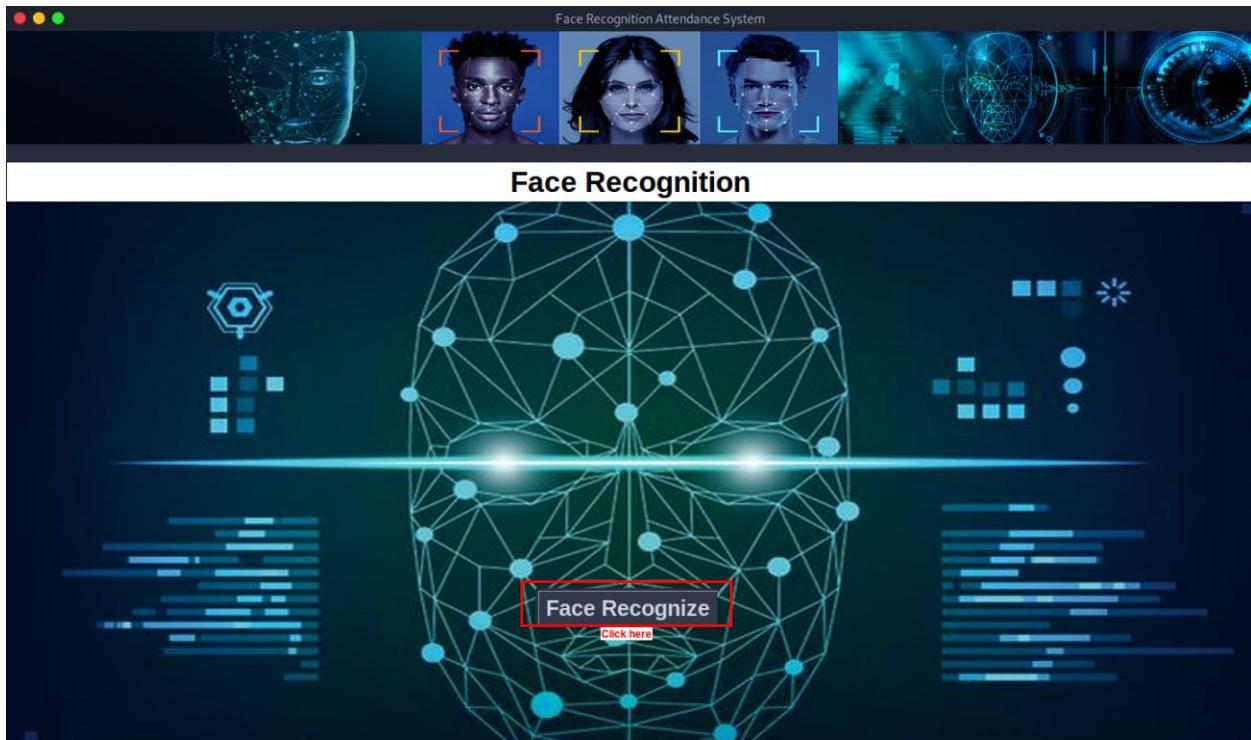


Figure 108: Error 03 (i)

```
#Image Recognition function
def recognize(img,clf,faceCascade):
    coord=draw_boundary(img,faceCascade,1,1,10,(255,25,255),"Face",clf)
    return img

faceCascade=cv2.CascadeClassifier("/home/hype/Downloads/FYP-Development/VS-Code/haarcascade_frontalface_default.xml")
clf=cv2.face.LBPHFaceRecognizer_create()
clf.read("/home/hype/Downloads/FYP-Development/VS-Code/Classifier.xml")

video_cap=cv2.VideoCapture(1) # Line 15

while True:
    ret,img=video_cap.read()
    img=recognize(img,clf,faceCascade)
    cv2.imshow("Welcome to FRAS",img)
    #cv2.waitKey(1)
    if cv2.waitKey(1)==13:
        break
video_cap.release()
cv2.destroyAllWindows()
```

VideoCapture Index  
not supporting  
camera model

Figure 109: Error 03 (ii)

**Error in code:** There was error occurred in code where VideoCapture index was not matching to the computer camera system.

**Solution to code:** Run camera.py for identifying index for VideoCapture.

```

1 import cv2
2 def return_camera_indices():
3     index = -10
4     arr = []
5     i = 10
6     while i > 0:
7         cap=cv2.VideoCapture(index)
8         if cap.read()[0]:
9             arr.append(index)
10        cap.release()
11        index+=1
12        i-=1
13    return arr
14 print(return_camera_indices())
return_camera_indices() > while i > 0

```

Run: camera

```

/home/hype/Downloads/FYP-Development/venv/bin/python /home/hype/Downloads/FYP-Development/VS-Code/camera.py
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1] ← Identifying VideoCapture supportive index to camera model

```

Process finished with exit code 0

Figure 110: Error 03 (iii)

```

#Image Recognition function
def recognize(img,clf,faceCascade):
    coord=draw_boundary(img,faceCascade,1.1,10,(255,25,255),"Face",clf)
    return img

faceCascade=cv2.CascadeClassifier("/home/hype/Downloads/FYP-Development/VS-Code/haarcascade_frontalface_default.xml")
clf=cv2.face.LBPHFaceRecognizer_create()
clf.read("/home/hype/Downloads/FYP-Development/VS-Code/Classifier.xml")

#Capture Video with index (-10,-9,-8,-7,-6,-5,-4,-3,-2,-1) to open camera// tested in camera module
video_cap=cv2.VideoCapture(-1) ← Using supporting VideoCapture Index

while True:
    ret,img=video_cap.read()
    img=recognize(img,clf,faceCascade)
    cv2.imshow("Welcome to FRAS",img)
    #cv2.waitKey(1)
    if cv2.waitKey(1)==13:
        break
video_cap.release()
cv2.destroyAllWindows()

```

Figure 111: Error 03 (iv)

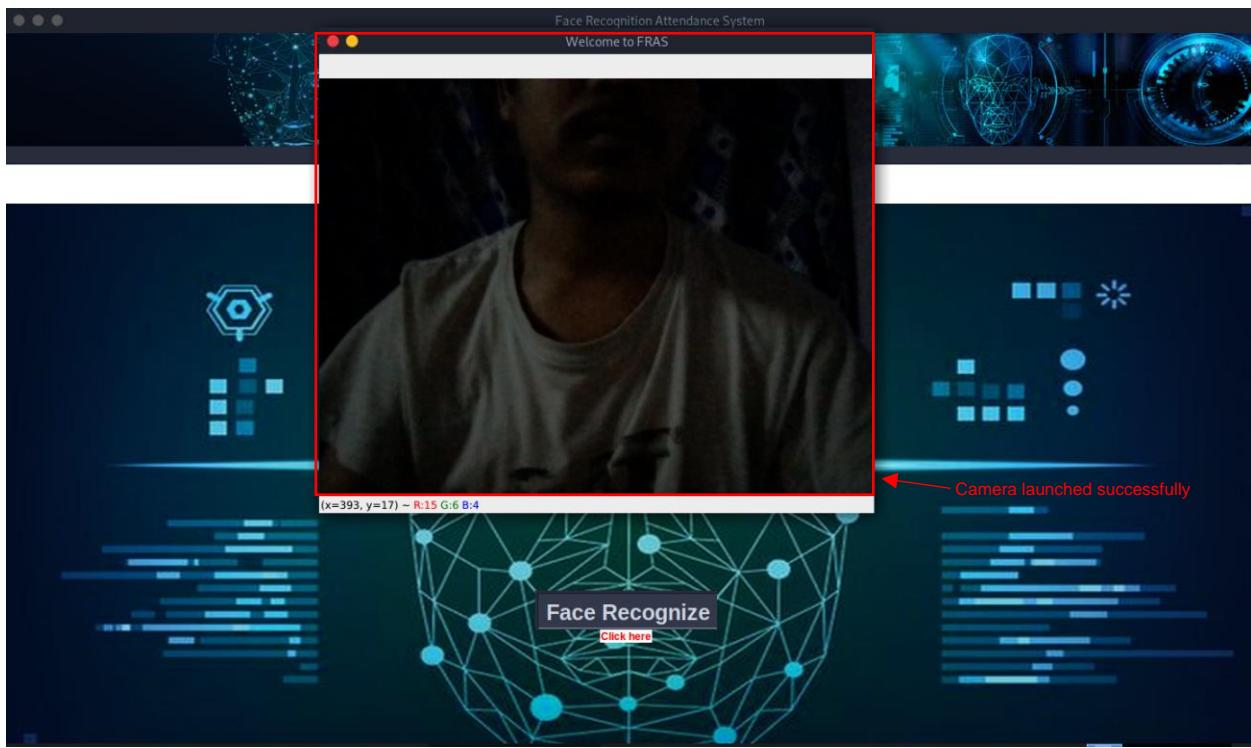


Figure 112: Error 03 (v)

**Solution Justification:**

Camera module can be launch through VideoCapture with specified index which varies from computer device and camera models. To identify the VideoCapture index, I wrote a program which returns the index which can launch camera module. In context of my camera model, index range from (-10) to (-1) is valid to launch camera.

## 4.4 Integration testing

### 4.4.1 Integration of database with the application

#### 4.4.1.1 Available Students

Student Detail							
Department	Course	Year	Semester	Student ID	Name	Division	Address
Information Technology	Networking	2020-202	Semester-2	1	DC	B	
Business	BBA	2022-202	Semester-1	1212	Pragyan Lama	C	

Figure 113: Available students (i)

Student Detail																
		Department	Course	Year	Semester	StudentID	Name	Division	Roll No	Gender	DOB	Email	Phone	Address	Teacher	
<input type="checkbox"/>		Edit		Copy		Delete	Information Technology	Networking	2020-2021	Semester-2	1	DC	B	1	Male	8/16/99 dc@gmail.com 9869791069 Parasi Prem Oli
<input type="checkbox"/>		Edit		Copy		Delete	Business	BBA	2022-2023	Semester-1	1212	Pragyan Lama	C	34	Male	4/5/02 praygan@gmail.com 9819293949 Bhairahawa Prem Oli

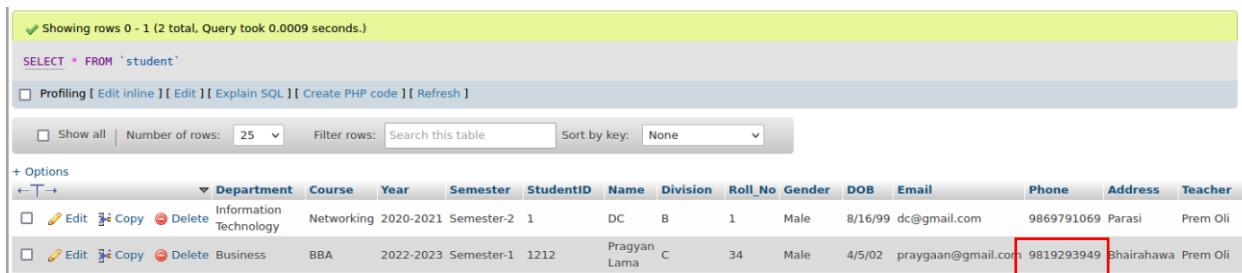
Figure 114: Available students (ii)

### 4.4.2 CRUD on Database

#### 4.4.2.1 Test 16: To edit a contact number of student in database

Test	16
Objective	To edit a contact number of student in database
Action	Clicked on ‘face_recognition’ database Clicked on ‘student table’ Clicked on edit button of one of student Change contact number to ‘9811995551’ Clicked ‘Go’ button
Expected Result	The contact number of student is supposed to change/update.
Actual Result	The contact number of student is successfully changed/updated.
Conclusion	Test successful.

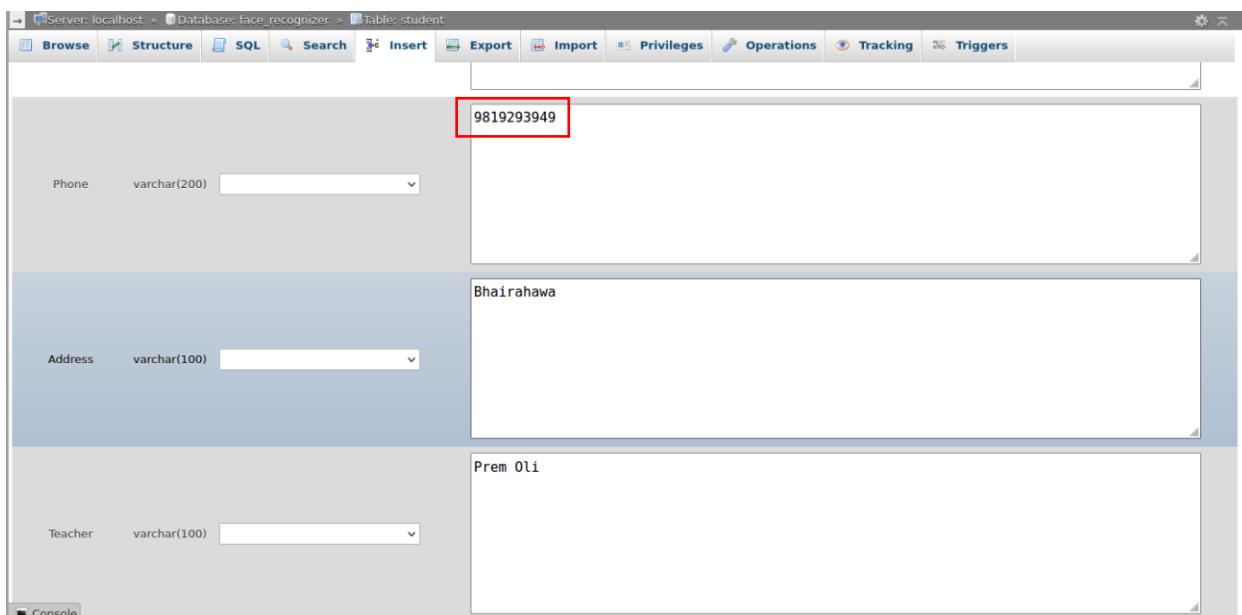
Table 19: Test 16: To edit a contact number of student in database



The screenshot shows the MySQL Workbench interface with the following details:

- Query Results:** Shows rows 0 - 1 (2 total). The query used was `SELECT * FROM 'student'`.
- Toolbar:** Includes options for Profiling, Edit inline, Explain SQL, Create PHP code, and Refresh.
- Table View:** Displays the student table with columns: Department, Course, Year, Semester, StudentID, Name, Division, Roll\_No, Gender, DOB, Email, Phone, Address, Teacher. Two rows are present:
  - Row 1: Department: Information Technology, Course: Networking, Year: 2020-2021, Semester: Semester-2, StudentID: 1, Name: DC, Division: B, Roll\_No: 1, Gender: Male, DOB: 8/16/99, Email: dc@gmail.com, Phone: 9869791069, Address: Parasi, Teacher: Prem Oli.
  - Row 2: Department: Business, Course: BBA, Year: 2022-2023, Semester: Semester-1, StudentID: 1212, Name: Pragyan Lama, Division: C, Roll\_No: 34, Gender: Male, DOB: 4/5/02, Email: praygaan@gmail.com, Phone: 9819293949, Address: Bhairahawa, Teacher: Prem Oli.

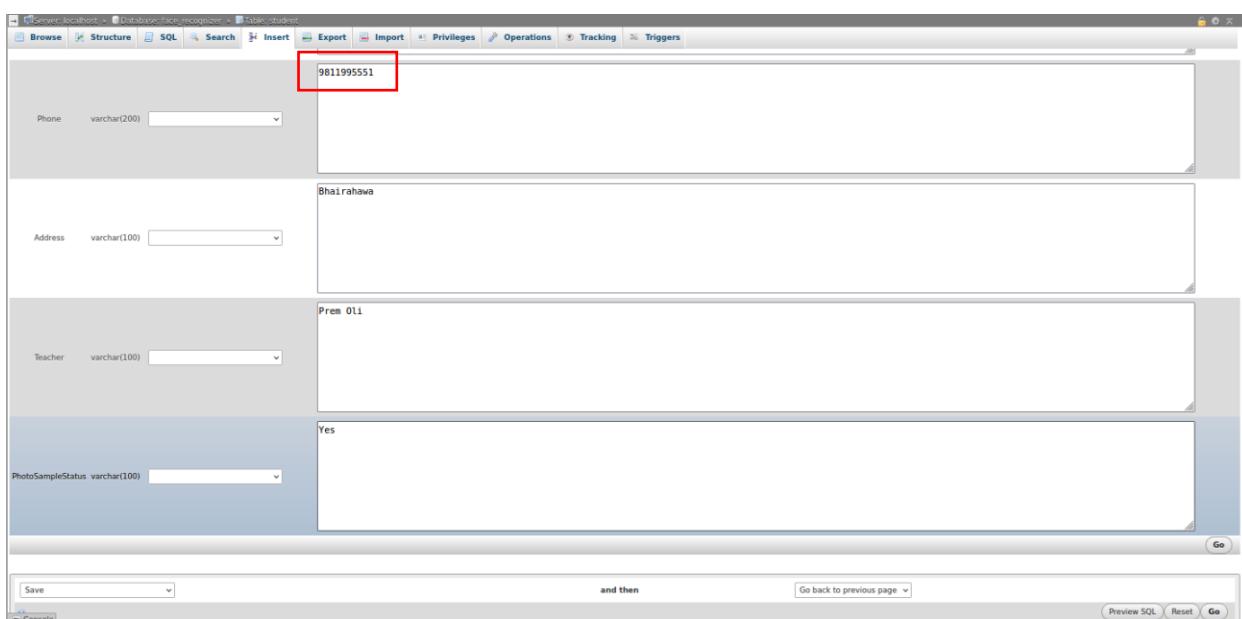
Figure 115: Test 16 (i)



The screenshot shows the MySQL Workbench interface with the following details:

- Navigation:** Server: localhost, Database: face\_recognizer, Table: student.
- Toolbar:** Includes Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, Triggers.
- Form Fields:**
  - Phone: varchar(200) - Value: 9819293949 (highlighted with a red box).
  - Address: varchar(100) - Value: Bhairahawa
  - Teacher: varchar(100) - Value: Prem Oli
- Console:** Shows the command `INSERT INTO student (Phone, Address, Teacher) VALUES ('9819293949', 'Bhairahawa', 'Prem Oli');`

Figure 116: Test 16 (ii)



The screenshot shows the MySQL Workbench interface with the following details:

- Navigation:** Server: localhost, Database: face\_recognizer, Table: student.
- Toolbar:** Includes Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, Triggers.
- Form Fields:**
  - Phone: varchar(200) - Value: 9811995551 (highlighted with a red box).
  - Address: varchar(100) - Value: Bhairahawa
  - Teacher: varchar(100) - Value: Prem Oli
  - PhotoSampleStatus: varchar(100) - Value: Yes
- Buttons:** Save, Go, Preview SQL, Reset, Go.
- Console:** Shows the command `INSERT INTO student (Phone, Address, Teacher, PhotoSampleStatus) VALUES ('9811995551', 'Bhairahawa', 'Prem Oli', 'Yes');`

Figure 117: Test 16 (iii)

Showing rows 0 - 1 (2 total. Query took 0.0018 seconds.)																
SELECT * FROM `student`																
Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]																
<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None										
+ Options	← T →	▼ Department	Course	Year	Semester	StudentID	Name	Division	Roll_No	Gender	DOB	Email	Phone	Address	Teacher	
<input type="checkbox"/>				Information Technology	Networking	2020-2021	Semester-2	1	DC	B	1	Male	8/16/99	dc@gmail.com	9869791069	Parasi Prem Oli
<input type="checkbox"/>				Business	BBA	2022-2023	Semester-1	1212	Pragyan Lama	C	34	Male	4/5/02	prayaan@gmail.com	9811995551	Bhairahawa Prem Oli

Figure 118: Test 16 (iv)

#### 4.4.2.2 Test 17: To delete one of the students

Test	17
Objective	To delete one of the students
Action	Clicked on ‘face_recognition’ database Clicked on ‘student table’ Clicked on ‘delete’ button to one of the student information Clicked ‘Ok button’
Expected Result	The student information is supposed to delete from database.
Actual Result	The student information is successfully deleted.
Conclusion	Test successful.

Table 20: Test 17: To delete one of the students

Showing rows 0 - 1 (2 total. Query took 0.0018 seconds.)																
SELECT * FROM `student`																
Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]																
<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None										
+ Options	← T →	▼ Department	Course	Year	Semester	StudentID	Name	Division	Roll_No	Gender	DOB	Email	Phone	Address	Teacher	
<input type="checkbox"/>				Information Technology	Networking	2020-2021	Semester-2	1	DC	B	1	Male	8/16/99	dc@gmail.com	9869791069	Parasi Prem Oli
<input type="checkbox"/>				Business	BBA	2022-2023	Semester-1	1212	Pragyan Lama	C	34	Male	4/5/02	prayaan@gmail.com	9811995551	Bhairahawa Prem Oli

Figure 119: Test 17 (i)

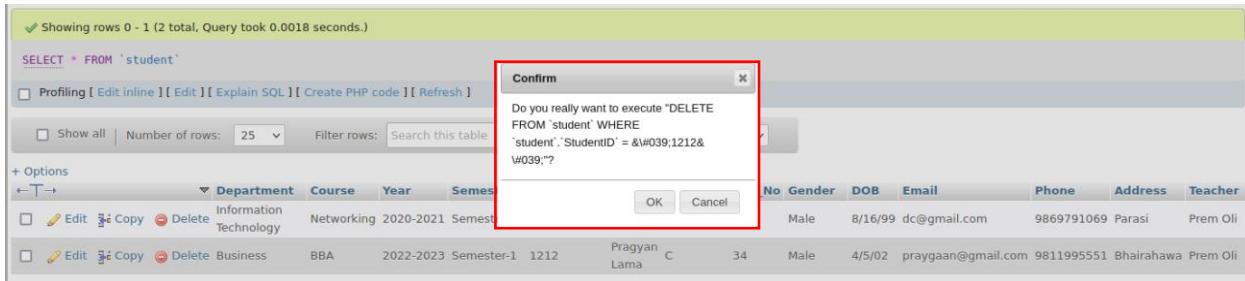


Figure 120: Test 17 (ii)

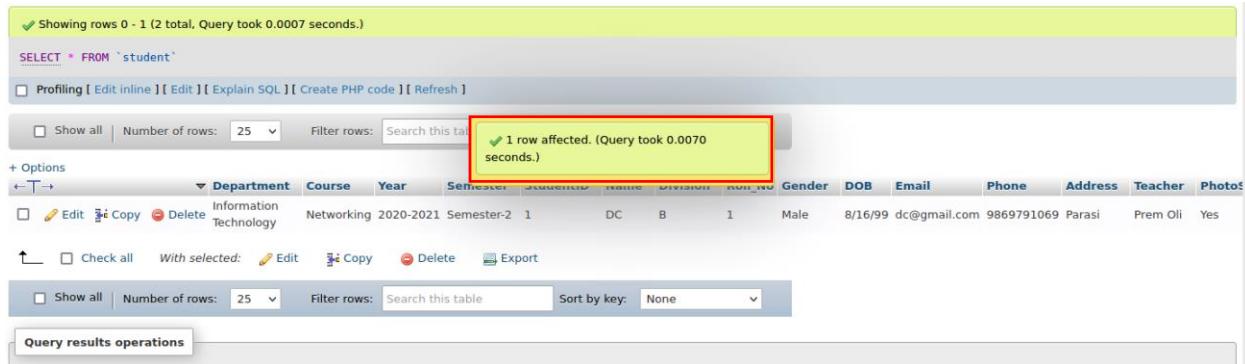


Figure 121: Test 17 (iii)

#### 4.4.2.3 Test 18: To edit security answer in database

Test	18
Objective	To edit security answer in database
Action	Clicked on 'face_recognition' database Clicked on 'register table' Clicked on 'edit' button to one of the teacher information Change security answer Clicked 'Go' button
Expected Result	The student information is supposed to delete from database.
Actual Result	The student information is successfully deleted.
Conclusion	Test successful.

Table 21: Test 18: To edit security answer in database

Form fields:

- Password varchar(100) [Input Field]
- ConfirmPassword varchar(100) [Input Field]
- SecurityQuestion varchar(100) [Input Field]
- SecurityAnswer varchar(50) [Input Field] (Value: jojo)

Buttons: Go, Save, and then, Go back to previous page, Preview SQL, Reset, Go.

Figure 122: Test 18 (i)

	<input type="checkbox"/> Edit	<input checked="" type="checkbox"/> Copy	<input type="checkbox"/> Delete	First Name	Last Name	Contact No.	Email	Password	ConfirmPassword	SecurityQuestion	SecurityAnswer
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sangam	Joshi	9823242526	joshi.sangam@gmail.com	2222222	2222222	Your pet name?	jojo
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Prem	Oli	9812223242	prem.olii52@gmail.com	Prem@53	Prem@52	Your pet name?	Sniffy

Figure 123: Test 18 (ii)

Form fields:

- Password varchar(100) [Input Field]
- ConfirmPassword varchar(100) [Input Field]
- SecurityQuestion varchar(100) [Input Field]
- SecurityAnswer varchar(50) [Input Field] (Value: Saggy)

Buttons: Go.

Figure 124: Test 18 (iii)

	<input type="checkbox"/> Edit	<input checked="" type="checkbox"/> Copy	<input type="checkbox"/> Delete	First Name	Last Name	Contact No.	Email	Password	ConfirmPassword	SecurityQuestion	SecurityAnswer
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sangam	Joshi	9823242526	joshi.sangam@gmail.com	2222222	2222222	Your pet name?	Saggy
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Prem	Oli	9812223242	prem.olii52@gmail.com	Prem@53	Prem@52	Your pet name?	Sniffy

Figure 125: Test 18 (iv)

#### **4.5 Critical Analysis**

The tests revealed that the system's features and capabilities all performed appropriately.

The results of the analysis performed following the testing are listed below:

- The overall functions and features of the system worked properly.
- The application was very convenient to use.
- The performance of the system was overall excellent.
- The UI of the application was excellent.
- A couple of validation had to be done in the forms.

## Chapter 5: Conclusion

### 5. Conclusion

Nowadays, various attendance and monitoring tools are used in practice in different places regardless of the fact that these solutions are mostly automatic, they are still prone to errors.

In this project, OpenCV, a Python open-source package used for computer vision in Artificial intelligence, Machine Learning, face recognition, and other areas, has been utilized to give HAAR-Cascade Detection and the LBPH algorithm for face identification. Using technology to eliminate faults not only saves resources, but it also decreases human intervention in the process by delegating all of the difficult tasks to storage. It is determined that with the required number of face images along with the proposed method of augmentation high accuracy can be determined.

To summarize, the system not only eliminates the problems that plague the old model of attendance, but it also makes it easier for the user to access the data collected by generating a CSV file.

#### 5.1 Legal, Social and Ethical Issues

In April 2021, the European Commission (EC) released its much-awaited Artificial Intelligence Act, a comprehensive regulatory proposal that classifies AI applications under distinct categories of risks. Among the identified high-risk applications, remote biometric systems, which include facial recognition technology (FRT), were singled out as particularly concerning. Their deployment, specifically in the field of law enforcement, may lead to human rights abuses in the absence of robust governance mechanisms (Lofred Madzou, 2021).

### 5.1.1 Legal Issues

Legal challenges arise when laws, rules and regulations are broken. Intellectual property ownership and intellectual property protection, such as trademarks, copyright, and patents, are some of the legal issues that applications confront nowadays. Terms and conditions of use, as well as privacy policies.

This IoT-based application is a simple but effective way to recognize faces and indicate attendance based on that recognition. As a result, none of these legal difficulties have been overlooked in the development of this application. This application maintains information secrecy and security and is legally recognized.

### 5.1.2 Social Issues

As the number of people utilizing the internet and technology has expanded and people are going digital, this application is aimed at the Nepalese audience and market. Given the number of people that use technology in their daily lives, the application has a beneficial social aspect. Because the initiative is beneficial to both the client and the company's students and staff. This program will help everyone because it saves people's time and paper. They can utilize this system for digitalized attendance, which eliminates the need for a paper register and saves time in today's fast-paced world. In every business and in every person's life, every second counts. As a result, this initiative and the entire system have no negative effects on society and are beneficial to future generations.

### 5.1.3 Ethical Issues

In recent years, critics questioned facial recognition systems' accuracy and role in identity fraud. Law enforcement agencies mistakenly implicated innocent people in riots in several cases. Additionally, identity management and storage remain questionable for many, haunting privacy advocates worldwide (Gangarapu, 2022). Now that more about the ethical

difficulties surrounding face recognition technology has been uncovered, let us examine it from a legal standpoint via three unique lenses that elicit criticism: discrimination, privacy, and democratic freedom. This technology protects the privacy of people's personal information. Personal information will not be available to the general public and will not be exploited.

## 5.2 Advantages

### 5.2.1 Automated time tracking system

Offices or workplaces or even just public places where the entry and exit times of employees or a person are strictly noted down will have a ready-made automated system to record the entry and exit time of each person for a given time. It won't even need the person to stop and click a photo, the software's are advanced enough to record the data from a continuous reel also. This means the flow won't get hampered, or you won't have to stop and smile or something like that. Just enter or exit the place effortlessly like you do every day and boom! Your attendance will be recorded without any fuss (Labs, 2020).

### 5.2.2 Cost-effective

Since the whole process will be done by a computer, it means the total attendance registration and calculation will be automated and done by the system itself, therefore, saving us the money which would have been otherwise spent on the labor cost to do that (Labs, 2020).

### 5.2.3 Time saving

Face recognition-based attendance system won't just calculate attendance but also note down the entry and exits of visitors in the place. At times when there is a situation where the identity and time of entry and exit of a specific person need to be noted, this system would become handy as it will easily show you when he/she came in and what are the places he/she

went to a very precise level. All of this means, you will have a much higher security level in your workplace (Labs, 2020).

#### **5.2.4 Increased security**

The whole world is suffering from COVID19 and it is high time we must give heed to social distancing. Having a safe distance with others has become a necessity nowadays. Times like this can be problematic if you have manual attendance system, having a Face recognition-based attendance system will not only allow you to register the attendance of a person but also keep you at a safe distance from them as you can work remotely and still see who all are coming and going. This calls for the point that, this whole system is a much safer, time-saving, and faster method to record attendance (Labs, 2020).

#### **5.2.5 Easy to manage**

Since the artificial intelligence-based attendance system is fully automated, managing the records and keeping a track of day to day activities will become much easier than the manual system. Everything will be done by the system. Many software is programmed in such a way that it shows the exact time of how many hours or minutes a person worked on his/her desk in the day. All this is can be done on a very large scale. Just imagine, recording the activities of a large crowd of 200 people simultaneously without any fuss and recording it at the same time in an organized manner! Such is the power of AI in face recognition (Labs, 2020).

### 5.3 Scope And Limitations

Face recognition, like any other technology, has potential downsides, such as risks to privacy, abuses of rights and personal freedoms, data theft, and other crimes. There's also the possibility of mistakes owing to technological problems. Although this method has several flaws, it has a lot of potential in today's environment. Here we talk about the project's scope and constraints.

#### 5.3.1 Scope of project

- This project's main purpose is to fix problems with the old attendance system while also recreating a brand new revolutionary smart system that will benefit the institution.
- Enables pupils' attendance to be tracked automatically.

#### 5.3.2 Limitation of project

- The biggest issue with face recognition is the considerable fluctuation of the recorded image owing to stance, lighting conditions, facial expression, various haircuts, the existence of glasses, and the presence of a beard.
- Difficulties in code writing
- Difficulty in overcome uncertainty.

#### 5.4 Future Works

Face Recognition Attendance System has an enormous future potential. It is flexible in terms of extension, so it may be changed as and when a need arises. Some characteristics can be tweaked further, such as the recognized distance, the usage of a graphics processing unit (GPU) for a big database, and the storing of speedy processing data on a server. It can also be integrated with numerous cameras at the same time.

However, due to a lack of time, this application was unable to include all of the anticipated functionality. This application could be improved even more, and new features could be added. The tasks that could be implemented in the future to develop and enhance the application are listed below:

- Using high-speed storage drivers and more RAM could result in a faster training time and response time for TensorFlow engine queries.
- Using a high-definition video camera capable of capturing live feed in 720p or 4K resolution can aid the constructed model in extracting high-level information.
- The application may use numerous cameras at the same time because it is highly coded.
- It can be connected to an online application using a server and a high-speed internet connection.
- If alternative GUI construction languages and libraries are used, the user interface of this application can be made friendlier.
- Many features could be added to this system, such as mailing attendance to the appropriate person, allowing parents to view attendance, and so on.

## Chapter 6: Reference

### 6. Reference

Patnaik, S. D. & T., 2020. *Student Attendance System using Face Recognition*. Trichy, India, Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020).

Narasimhan, T. E., 2019. *How facial recognition technology is transforming attendance marking system*. Chennai: Business Standard.

Thales, 2021. *Facial recognition*. [Online]

Available at: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>

[Accessed 7 Dec 2021].

McClelland, C., 2020. *What is an IoT Platform?*. Rockville, Maryland: IoT for all.

Link Labs, 2016. *IoT Platforms: What They Are & How To Select One*. [Online]

Available at: <https://www.link-labs.com/blog/what-is-an-iot-platform>

[Accessed 07 Dec 2021].

Arpankumar Patel, A. J. S. S. R. N., 2019. *Smart Student Attendance System Using QR Code*, Mumbai: KJ Somaiya Inst of Engineering & Information Tech.

G.T.Bharathy, S. T., 2021. Smart Attendance Monitoring System using IoT and RFID. *Internaltional Journal of Advances in Engineering and Management*, 3(6 June 2021), pp. 1307-1313.

Muralidharan P, V. A., 2016. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(6).

Martinez, P., 2020. *Evolutionary Prototyping*. [Online]

Available at: <https://mockitt.wondershare.com/prototyping/evolutionary-prototyping.html>  
[Accessed 24 Nov 2021].

Ronak Dalvi, A. S. R. B. B. R. P. & M. K. S., Apr 2019. Biometric Attendance System. *International Research Journal of Engineering and Technology (IRJET)*, 06(04), p. 4110.

McCarthy, D., 2005. Product Review: XAMPP, an Apache Distribution. *Linux Journal*.

Dwivedi, R., 2020. Pycharm IDE For Dummies- Beginners Guide. *DEVELOPERS CORNER*, 02 Aug.

Amos, D., 2022. *Python GUI Programming With Tkinter*. [Online]

Available at: <https://realpython.com/python-gui-tkinter/#:~:text=Python%20has%20a%20lot%20of,Windows%2C%20macOS%2C%20a>

nd%20Linux.

[Accessed 3 April 2022].

Python Software Foundation , 2022. *What is Python? Executive Summary*. [Online]  
Available at: <https://www.python.org/doc/essays/blurb/>  
[Accessed 03 April 2022].

Python software foundation, 2022. *Pillow*. [Online]  
Available at: <https://pypi.org/project/Pillow/>  
[Accessed 03 April 2022].

Oracle Corporation, 2022. *Python MySQL Connector*. [Online]  
Available at: <https://dev.mysql.com/doc/connector-python/en/connector-python-introduction.html>  
[Accessed 03 April 2022].

Venkatesh, B., 2020. *WHAT IS THE OPENCV LIBRARY AND WHY DO WE NEED TO KNOW ABOUT IT?*. [Online]  
Available at: <https://www.topcoder.com/thrive/articles/what-is-the-opencv-library-and-why-do-we-need-to-know-about-it>  
[Accessed 03 April 2022].

Martin, Z., 2015. *Facial recognition on the rise, FIDO adds support for Bluetooth and NFC*. [Online]  
Available at: <https://www.secureidnews.com/news-item/facial-recognition-on-the-rise-fido-adds-support-for-bluetooth-and-nfc/>  
[Accessed 04 March 2022].

Elbehiery, H., 2019. Enhancement of QR code Student's Attendance Management System using GPS. *IOSR Journal of Computer Engineering*, 21(4), pp. 18-30.

taifur, 2018. *Smart Attendance System (Intel Edison Inside)*. [Online]  
Available at: <https://www.instructables.com/Smart-Attendance-System-Intel-Edison-Inside/>  
[Accessed 17 March 2022].

up tech, 2020. *8 Best Software Development Methodologies*. [Online]  
Available at: <https://www.uptech.team/blog/software-development-methodologies>  
[Accessed 19 March 2022].

Tutorial Teacher, 2020. *Python - OS Module*. [Online]  
Available at: <https://www.tutorialsteacher.com/python/os-module>  
[Accessed 04 March 2022].

Python Software Foundation, 2022. *CSV*. [Online]  
Available at: <https://docs.python.org/3/library/csv.html>  
[Accessed 05 March 2022].

Pyhton Software Foundation, 2022. *datetime*. [Online]  
Available at: <https://docs.python.org/3/library/datetime.html>  
[Accessed 05 March 2022].

Python Software Foundation, 2022. *time*. [Online]  
Available at: <https://docs.python.org/3/library/time.html>  
[Accessed 05 March 2022].

Python Software Foundation, 2022. *tkcalendar 1.6.1*. [Online]  
Available at: <https://pypi.org/project/tkcalendar/>  
[Accessed 05 March 2022].

International Journal of Innovative Research in Science, Engineering and Technology, 2019. Real-time Facial Recognition using Haar Cascade. *International Journal of Innovative Research in Science, Engineering and Technology*, 8(11), pp. 10832-10834.

International Journal of Innovative Research in Science, E. a. T., 2019. Real-time Facial Recognition using Haar Cascade. *International Journal of Innovative Research in Science, Engineering and Technology*, 8(11), pp. 10834-10835.

Lofred Madzou, J. R. S. L. L. G. O. M. M. E., 2021. *This is best practice for using facial recognition in law enforcement*. [Online]  
Available at: <https://www.weforum.org/agenda/2021/10/facial-recognition-technology-law-enforcement-human-rights/>  
[Accessed 12 March 2022].

Gangarapu, K. R., 2022. *Ethics of Facial Recognition: Key Issues and Solutions*. [Online]  
Available at: <https://learn.q2.com/ethics-of-facial-recognition#:~:text=The%20top%20six%20ethical%20concerns,breaches%2C%20and%20inefficient%20legal%20support.>  
[Accessed 12 March 2022].

Imperva, 2021. *White Box Testing*. [Online]  
Available at: <https://www.imperva.com/learn/application-security/white-box-testing/>  
[Accessed 02 April 2022].

Imperva, 2021. *What is Black Box Testing*. [Online]  
Available at: <https://www.imperva.com/learn/application-security/black-box-testing/>  
[Accessed 03 April 2022].

Hamilton, T., 2022. *Integration Testing*. [Online]  
Available at: <https://www.guru99.com/integration-testing.html>  
[Accessed 16 April 2022].

Labs, A., 2020. *Face Recognition Based Attendance*. [Online]  
Available at: <https://www.aindralabs.com/face-recognition-based-attendance->

advantages-disadvantages-and-the-major-concerns/  
[Accessed 04 April 2022].

## Chapter 7: Bibliography

Patnaik, S. D. & T., 2020. *Student Attendance System using Face Recognition*. Trichy, India, Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020).

Narasimhan, T. E., 2019. *How facial recognition technology is transforming attendance marking system*. Chennai: Business Standard.

Thales, 2021. *Facial recognition*. [Online]

Available at: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>

[Accessed 7 Dec 2021].

McClelland, C., 2020. *What is an IoT Platform?*. Rockville, Maryland: IoT for all.

Link Labs, 2016. *IoT Platforms: What They Are & How To Select One*. [Online]

Available at: <https://www.link-labs.com/blog/what-is-an-iot-platform>

[Accessed 07 Dec 2021].

Arpankumar Patel, A. J. S. S. R. N., 2019. *Smart Student Attendance System Using QR Code*, Mumbai: KJ Somaiya Inst of Engineering & Information Tech.

G.T.Bharathy, S. T., 2021. Smart Attendance Monitoring System using IoT and RFID. *International Journal of Advances in Engineering and Management*, 3(6 June 2021), pp. 1307-1313.

Muralidharan P, V. A., 2016. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(6).

Martinez, P., 2020. *Evolutionary Prototyping*. [Online]

Available at: <https://mockitt.wondershare.com/prototyping/evolutionary-prototyping.html>  
[Accessed 24 Nov 2021].

Ronak Dalvi, A. S. R. B. B. R. P. & M. K. S., Apr 2019. Biometric Attendance System. *International Research Journal of Engineering and Technology (IRJET)*, 06(04), p. 4110.

McCarthy, D., 2005. Product Review: XAMPP, an Apache Distribution. *Linux Journal*.

- Dwivedi, R., 2020. Pycharm IDE For Dummies- Beginners Guide. *DEVELOPERS CORNER*, 02 Aug.
- Amos, D., 2022. *Python GUI Programming With Tkinter*. [Online] Available at: <https://realpython.com/python-gui-tkinter/#:~:text=Python%20has%20a%20lot%20of,Windows%2C%20macOS%2C%20and%20Linux>. [Accessed 3 April 2022].
- Python Software Foundation , 2022. *What is Python? Executive Summary*. [Online] Available at: <https://www.python.org/doc/essays/blurb/> [Accessed 03 April 2022].
- Python software foundation, 2022. *Pillow*. [Online] Available at: <https://pypi.org/project/Pillow/> [Accessed 03 April 2022].
- Oracle Corporation, 2022. *Python MySQL Connector*. [Online] Available at: <https://dev.mysql.com/doc/connector-python/en/connector-python-introduction.html> [Accessed 03 April 2022].
- Venkatesh, B., 2020. *WHAT IS THE OPENCV LIBRARY AND WHY DO WE NEED TO KNOW ABOUT IT?*. [Online] Available at: <https://www.topcoder.com/thrive/articles/what-is-the-opencv-library-and-why-do-we-need-to-know-about-it> [Accessed 03 April 2022].
- Martin, Z., 2015. *Facial recognition on the rise, FIDO adds support for Bluetooth and NFC*. [Online] Available at: <https://www.secureidnews.com/news-item/facial-recognition-on-the-rise-fido-adds-support-for-bluetooth-and-nfc/> [Accessed 04 March 2022].
- Elbehery, H., 2019. Enhancement of QR code Student's Attendance Management System using GPS. *IOSR Journal of Computer Engineering*, 21(4), pp. 18-30.
- taifur, 2018. *Smart Attendance System (Intel Edison Inside)*. [Online] Available at: <https://www.instructables.com/Smart-Attendance-System-Intel-Edison-Inside/> [Accessed 17 March 2022].
- up tech, 2020. *8 Best Software Development Methodologies*. [Online] Available at: <https://www.uptech.team/blog/software-development-methodologies> [Accessed 19 March 2022].

Tutorial Teacher, 2020. *Python - OS Module*. [Online]  
Available at: <https://www.tutorialsteacher.com/python/os-module>  
[Accessed 04 March 2022].

Python Software Foundation, 2022. *CSV*. [Online]  
Available at: <https://docs.python.org/3/library/csv.html>  
[Accessed 05 March 2022].

Pyhton Software Foundation, 2022. *datetime*. [Online]  
Available at: <https://docs.python.org/3/library/datetime.html>  
[Accessed 05 March 2022].

Python Software Foundation, 2022. *time*. [Online]  
Available at: <https://docs.python.org/3/library/time.html>  
[Accessed 05 March 2022].

Python Software Foundation, 2022. *tkcalendar 1.6.1*. [Online]  
Available at: <https://pypi.org/project/tkcalendar/>  
[Accessed 05 March 2022].

International Journal of Innovative Research in Science, Engineering and Technology, 2019. Real-time Facial Recognition using Haar Cascade. *International Journal of Innovative Research in Science, Engineering and Technology*, 8(11), pp. 10832-10834.

International Journal of Innovative Research in Science, E. a. T., 2019. Real-time Facial Recognition using Haar Cascade. *International Journal of Innovative Research in Science, Engineering and Technology*, 8(11), pp. 10834-10835.

Lofred Madzou, J. R. S. L. L. G. O. M. M. E., 2021. *This is best practice for using facial recognition in law enforcement*. [Online]  
Available at: <https://www.weforum.org/agenda/2021/10/facial-recognition-technology-law-enforcement-human-rights/>  
[Accessed 12 March 2022].

Gangarapu, K. R., 2022. *Ethics of Facial Recognition: Key Issues and Solutions*. [Online]  
Available at: <https://learn.g2.com/ethics-of-facial-recognition#:~:text=The%20top%20six%20ethical%20concerns,breaches%2C%20and%20inefficient%20legal%20support.>  
[Accessed 12 March 2022].

Imperva, 2021. *White Box Testing*. [Online]  
Available at: <https://www.imperva.com/learn/application-security/white-box-testing/>  
[Accessed 02 April 2022].

Imperva, 2021. *What is Black Box Testing*. [Online]  
Available at: <https://www.imperva.com/learn/application-security/black-box-testing/>  
[Accessed 03 April 2022].

Hamilton, T., 2022. *Integration Testing*. [Online]  
Available at: <https://www.guru99.com/integration-testing.html>  
[Accessed 16 April 2022].

Labs, A., 2020. *Face Recognition Based Attendance*. [Online]  
Available at: <https://www.aindralabs.com/face-recognition-based-attendance-advantages-disadvantages-and-the-major-concerns/>  
[Accessed 04 April 2022].

## Chapter 8: Appendix

### 7. Appendix

#### 8.1 Considered Methodology

##### 8.1.1 Agile Development Methodology

Agile methodology is arguably one of the most popular software development methodologies in recent days. It takes a different approach from the conventional, linear method. Agile focuses on how to satisfy the users instead of emphasizing documentation and rigid procedures (up tech, 2020).

#### Pros

- Software has minimal defects due to the iterative effort in testing and fine-tuning.
- Clarity between team members during development, thanks to the frequent and transparent development.
- Changes in project requirements are easily addressed and work on with little impact on the timeline.
- An overall improvement on the deliverable's quality.

#### Cons

- The team can sometimes lose focus due to overwhelming change requests.
- Documentation takes a back seat in Agile, which can be a problem later on in development.
- Agile focuses on discussions and feedback, which can be too time-consuming for the team.
- Due to its non-structured approach, Agile requires experienced developers who can work independently.

### 8.1.2 Waterfall Development Methodology

Despite decades since it was first used, the waterfall methodology is still relevant in some projects today. It is a simple, linear method where development stages are arranged into sequential, cascading processes. The waterfall development methodology is easily understood, which makes it popular for teams with lesser design experience. Each stage must be completed before moving on to the next. For example, all the requirements must be established before design can commence (up tech, 2020).

#### Pros

- The linearity of the waterfall model makes it easy to understand, particularly for new developers.
- All specifications and deliverables are spelled out before the development commence.
- There's no room for miscommunicating information with the waterfall model as it's clearly defined in each stage.

#### Cons

- It doesn't include customer feedback in the early phases, which increases the risk of the project veering off target.
- Testing is only executed at the end of the development. Some problems are harder to fix at a later stage.
- The rigidity of the waterfall model gives no room for changes, making it unsuitable for complex projects.

- The team can spend too much time on documentation instead of delivering solutions that solve the user's problems.

### 8.1.3 Prototype Model

Instead of developing a full-fledged software, the prototype model allows developers to work on the prototype version of the final product. The prototype is then made available for customer testing, evaluation, and feedback. Based on the gathered feedback, the prototype goes through several iterations of refinement until it's deemed satisfactory by the customer. This prototype is categorized into multiple variation with its unique features. The appeal of the prototype approach is its rigorous evaluation that uncovers possible issues before actual development begins (up tech, 2020).

#### Pros

- Good in ironing out potential issues in the early development stage, which greatly reduces product failure risk.
- Ensures the customer is happy with the 'product', before real development works started.
- Build rapport with the customer early on with the discussions, which helps throughout the project.
- Gather detailed information with the prototype, which is later on used in building the final version.

#### Cons

- Excessive to and fro in testing out the prototype with the customer can delay the development timeline.
- The customer's expectations of the actual product may not align with the prototype.

- There's a risk of cost overrun as the works on the prototype are often paid for by the developer.

## 8.2 Appendix B: Python Libraries

### 8.2.1 Pillow

Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors. As of 2019, Pillow development is supported by Tidelift. The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool (Python software foundation, 2022).

- **Installation-** \$ pip install pillow



Figure 126: Logo of Pillow Library

### 8.2.2 MYSQL connector

MySQL Connector is a library that communicate with MySQL database servers. MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249) (Oracle Corporation, 2022).

- **Installation-** \$ pip install mysql-connector-python

### 8.2.3 OpenCV

Opencv is an open-source library which is very useful for computer vision applications such as video analysis, CCTV footage analysis and image analysis. OpenCV is written by C++ and has more than 2,500 optimized algorithms. When we create applications for computer vision that we don't want to build from scratch we can use this library to start focusing on real world problems. There are many companies using this library today such as Google, Amazon, Microsoft and Toyota. Many researchers and developers contribute. We can easily install it in any OS like Windows, Ubuntu and MacOS (Venkatesh, 2020).

- **Installation-** \$ pip install opencv-python

### 8.2.4 OS

The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc. You first need to import the os module to interact with the underlying operating system. So, import it using the import os statement before using its functions (Tutorial Teacher, 2020).

### 8.2.5 CSV

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. The csv module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe

the CSV formats understood by other applications or define their own special-purpose CSV formats (Python Software Foundation, 2022).

### 8.2.6 Datetime

The `datetime` module supplies classes for manipulating dates and times. Date and time objects may be categorized as “aware” or “naive” depending on whether or not they include time zone information (Pyhton Software Foundation, 2022).

### 8.2.7 Time

This module provides various time-related functions. For related functionality, see also the `datetime` and `calendar` modules. Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms (Python Software Foundation, 2022).

### 8.2.8 TKcalendar

`Tkcalendar` is a python module that provides the `Calendar` and `DateEntry` widgets for Tkinter. The `DateEntry` widget is similar to a Combo box, but the drop-down is not a list but a Calendar to select a date. Events can be displayed in the `Calendar` with custom colors and a tooltip displays the event list for a given day. `tkcalendar` is compatible with both Python 2 and Python 3. It supports many locale settings (e.g. ‘fr\_FR’, ‘en\_US’, ..) and the colors are customizable (Python Software Foundation, 2022).

### 8.3 Appendix C: Function and Feature

The application's functionality and features are based on the specifications provided by the client. The system also includes additional developer and helpdesk information. The following characteristics are required:

#### 8.3.1 Register

User/teacher can register an account.

#### 8.3.2 Login

Using valid email and password, user can login into the system.

#### 8.3.3 Forget password

User can request for reset password in case of forgotten password.

#### 8.3.4 modify student detail

User can add, remove or update student detail.

#### 8.3.5 Take student photo

User can take student photo while adding a new student and later can also update photo into the system.

#### 8.3.6 Generate dataset

User can generate dataset and its automatically get save into the database.

#### 8.3.7 Train dataset

User can train the dataset on a single click which take less than 5 minutes.

#### 8.3.8 Modify attendance

User can modify attendance, delete or update the student attendance.

#### 8.3.9 Import/export attendance sheet

User can import and export the csv file (attendance file).

## 8.4 Appendix C: Pre-Survey

### 8.4.1 Pre-Survey Form



The form features a decorative header image showing three people (two adults and one child) watching a vibrant fireworks display over water at night.

## Face Recognition Attendance System

Survey for implementation of 'Face Recognition Attendance System'

**Email \*** [np01nt4a190062@islingtoncollege.edu.np](mailto:np01nt4a190062@islingtoncollege.edu.np) [Switch accounts](#) 

**Name \***

Your answer

Are you interested in Internet of Things (IoT)? \*

Yes

No

Maybe

Figure 127: Pre-Survey form (i)

Do you have any idea of Biometric Attendance system?

Yes  
 No  
 Not Sure

Did you ever use any of Biometric Attendance system yet?

Yes  
 No  
 Not Sure

Which attendance system do you prefer most? \*

Tradition Attendance system(Paperwork)  
 Biometric Attendance system

Which Biometric technology may be best for Attendance system? \*

Face Recognition  
 Fingerprint  
 Voice Recognition  
 Iris Recognition  
 Retina Scan  
 Keystroke Dynamics  
 Signature Recognition  
 None of above

Figure 128: Pre-Survey form (ii)

Do you think Face Recognition Attendance system has been used by any organization/institute yet? \*

- Yes
- No
- May be

Do you recommend to use this system to others? \*

- Yes
- No
- May be

In your opinion, how feasible is face recognition attendance system in context of Nepal? \*

- High
- Medium
- Low

Is it favorable in the present condition of Coronavirus Pandemic? \*

- Yes
- No
- Maybe

Figure 129: Pre-Survey form (iii)

Do you think this project eliminate problem related to traditional attendance system(paper-pen attendance)? \*

- Yes
- No
- Maybe

What additional feature do you wish in this system?

Your answer

---

Please, do rate this system \*

1      2      3      4      5      6      7      8      9      10

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 

Any feedback or suggestion regarding this project?

Your answer

---

**Submit**

**Clear form**

Figure 130: Pre-Survey form (iv)

#### 8.4.2 Sample of Filled Pre-Survey Forms

The screenshot shows a Google Forms interface with the following details:

- Responses:** 21 responses
- Status:** Accepting responses (switch is green)
- Navigation:** Summary (selected), Question, Individual
- Section Title:** Who has responded?
- Emails listed:**
  - sapkotaalisha43@gmail.com
  - anilbadayak7@gmail.com
  - khemkumarichaudhary02@gmail.com
  - gautambarun20@gmail.com
  - kedemlimbuana@gmail.com
  - aakashshahu59@gmail.com
  - seraph446@gmail.com
  - np01nt4a190181@islingtoncollege.edu.np

Figure 131: Pre-Survey Response (i)

Who has responded?
bhola.chryy@gmail.com
rojin.dumre98@gmail.com
np01nt4a190106@islingtoncollege.edu.np
rabikumarshah21@gmail.com
santos.dc101@gmail.com
be.hype069@gmail.com
ashishbhurtel123@gmail.com
karanchy227@gmail.com
ybinod857@gmail.com

Figure 132: Pre-Survey Response (ii)

bistakapur20@gmail.com
sgrcarkey@gmail.com
sureshbudha2@gmail.com
creativeinfotechsolution683@gmail.com

Figure 133: Pre-Survey Response (iii)

Name

21 responses

Anil

Bhola Chaudhary

Suresh Budha

Khem Kumari chaudhary

Seraph Shrestha

Rabi Kumar Shah

Sagar Karki

Safal Karki

Kapur Bista

Figure 134: Pre-Survey Response (iv)

Name  
21 responses

anamol limbu
Binod Ray
Alisha Sapkota
Karan Chaudhary
Ashish Bhurtel
Rojin Dumre
Santosh chaudhary
Binod Ray
Creative InfoTech Solution Private Limited

Figure 135: Pre-Survey Response (v)

Aakash Sah
Barun Gautam
Nirajan Sharma

Figure 136: Pre-Survey Response (vi)

Are you interested in Internet of Things (IoT)?

19 responses

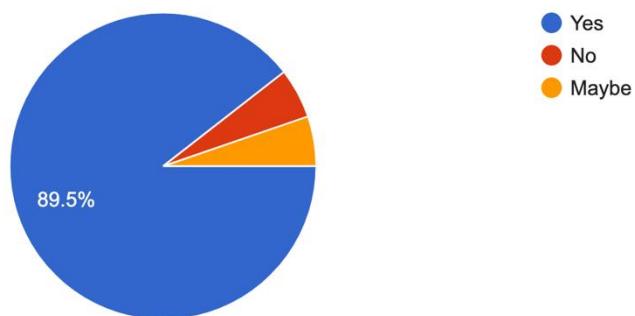


Figure 137: Pre-Survey Response (vi)

Do you have any idea of Biometric Attendance system?

20 responses

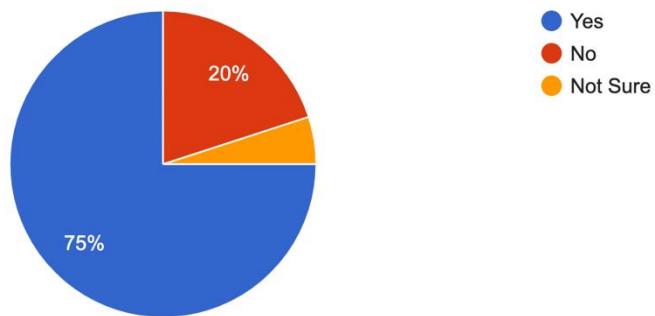


Figure 138: Pre-Survey Response (vii)

Did you ever use any of Biometric Attendance system yet?

21 responses

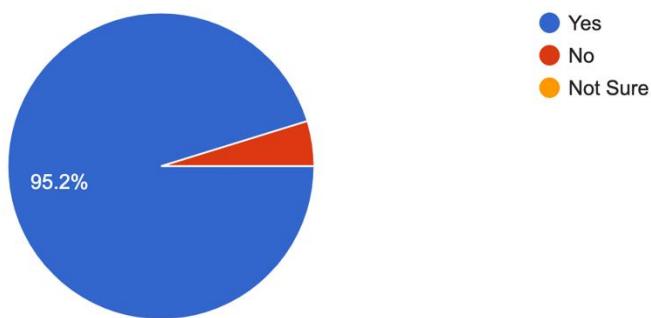


Figure 139: Pre-Survey Response (viii)

Which attendance system do you prefer most?

19 responses

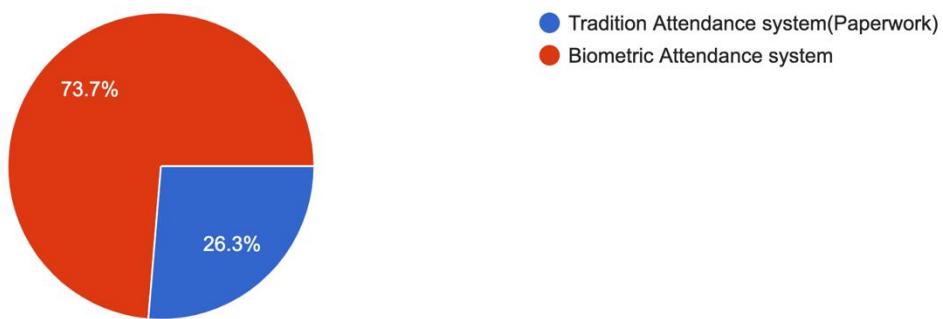


Figure 140: Pre-Survey Response (ix)

Which Biometric technology may be best for Attendance system?

21 responses

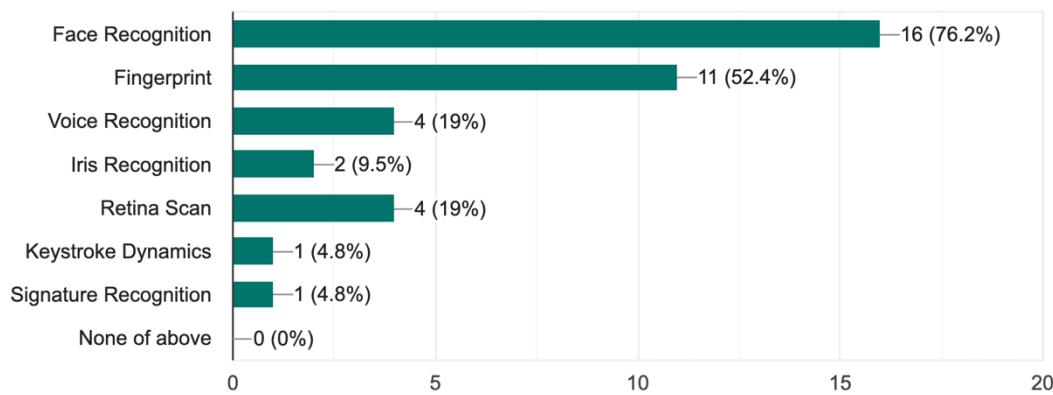


Figure 141: Pre-Survey Response (x)

Do you think face recognition attendance system can overcome other biometric system?

21 responses

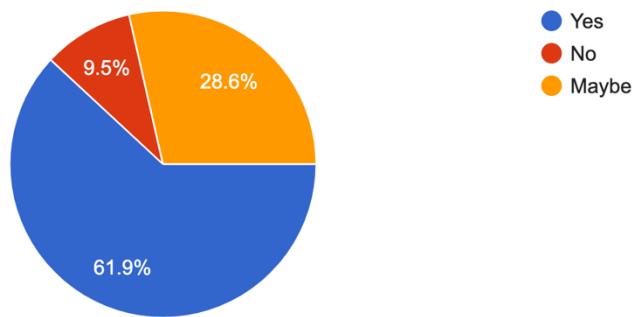


Figure 142: Pre-Survey Response (xi)

Do you think Face Recognition Attendance system has been used by any organization/institute yet?  
21 responses

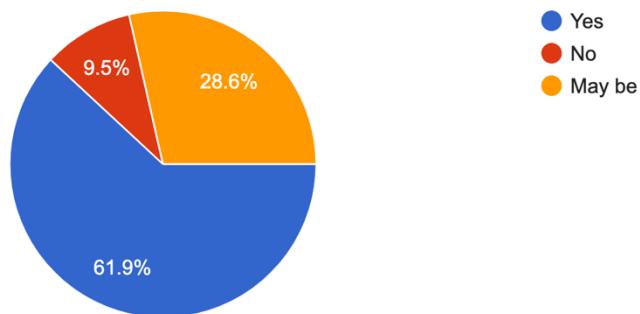


Figure 143: Pre-Survey Response (xii)

Do you recommend to use this system to others?  
19 responses

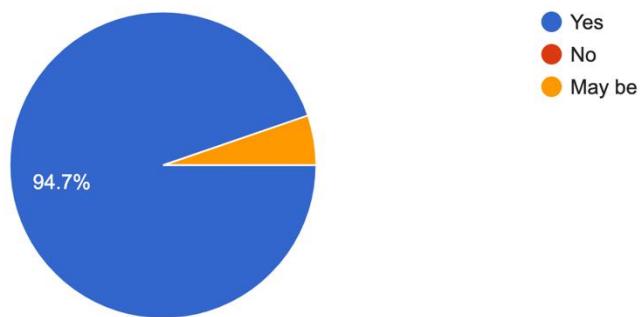


Figure 144: Pre-Survey Response (xiii)

In your opinion, how feasible is face recognition attendance system in context of Nepal?  
21 responses

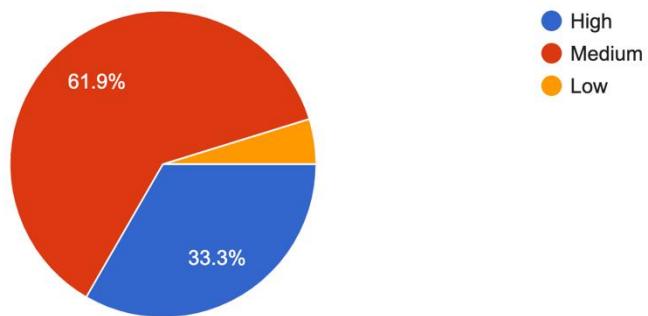


Figure 145: Pre-Survey Response (xiv)

Is it favorable in the present condition of Coronavirus Pandemic?  
19 responses

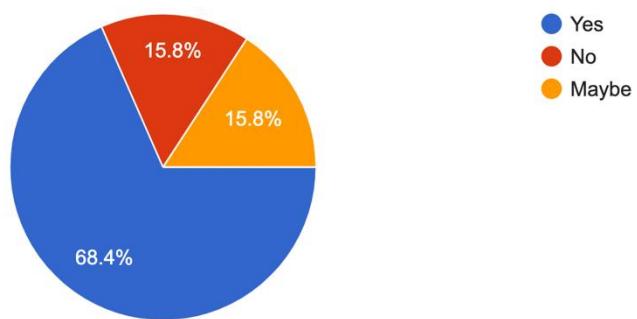


Figure 146: Pre-Survey Response (xv)

Do you think this project eliminate problem related to traditional attendance system(paper-pen attendance)?

16 responses

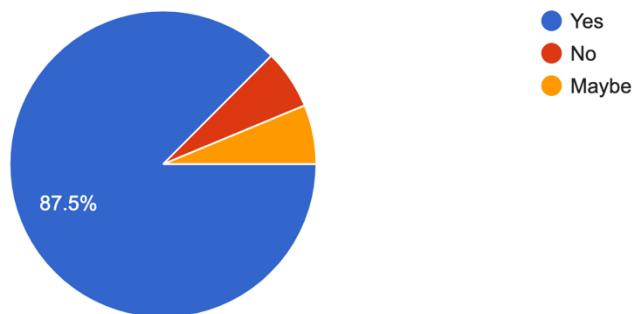


Figure 147: Pre-Survey Response (xvi)

What additional feature do you wish in this system?

8 responses

Biometric is better

Score chart showing attendance rate .

The system should say if you are ugly or pretty

It would be really good if the system showed other details from face recognition like temperature and things like that

any sort of alert system may go good

System should include mail system to those who are absent or present both.

Face Recognition with mask alert

auto mail may be good for present or absent

Please, do rate this system

 Copy

19 responses

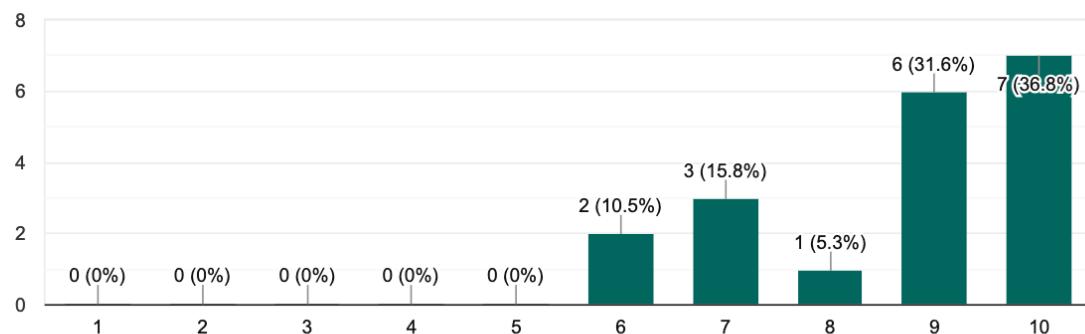


Figure 148: Pre-Survey Response (xvii)

Any feedback or suggestion regarding this project?

6 responses

Do not have

Add a score chart showing attendance rate .

its nice tara retina select garna milya chaina work on that

This project is a huge undertaking that has been progressing duly with great effort of Dharmaraj. I think this project has the capacity to be successful in the real world if implemented properly. Overall, I think this project will be successful in most work environments.

Being client, this project might be much helpful for quick time attendance for out employees and student.

No

Figure 149: Pre-Survey Response (xviii)

#### 8.4.3 Pre-Survey Result

- Total 21 peoples had completed survey with their opinions and choices.
- 89.5% of them are interested in IoT projects where as 5.3% are not interested at all.
- 75% of peoples had idea of biometric attendance previously whereas 20% of them are still unaware of biometric attendance.
- Most of them seems used biometric attendance which is 95.2% of them.
- It seems like 73.7% of people prefer biometric attendance system rather than traditional attendance system.
- Among all the biometric system, face recognition seems best of all as 76.2% of people prefers it.
- 61.9% people thinks that face recognition system might overcome other biometric system.
- 61.9% of all thinks that many of the companies are already using biometric attendance system.
- 94.7% of them prefer to recommend to use this system.
- 61.9% of them thinks it might be feasible in context of Nepal.
- 15.8% of them thinks it might not be favourable in present condition of pandemic.
- Almost 38% of them rated this system 10 out of 10.

## 8.5 Appendix B: Post-Survey

### 8.5.1 Post-Survey Form



The form features a decorative header image showing three people (two adults and one child) watching a vibrant display of fireworks against a dark sky.

## Face Recognition Attendance System(FRAS)

A contactless facial recognition attendance system eliminates any physical interaction between the man and the machine. Knowing how the technology works makes it much easier to comprehend how facial recognition attendance systems may make buildings and premises safer and more efficient.

np01nt4a190062@islingtoncollege.edu.np [Switch accounts](#) 

\*Required

Email \*

Your email address

Name \*

Your answer

Have you ever used Face Recognition Attendance system? \*

Yes

No

Not Sure

Figure 150: Post-Survey Form (i)

If it is applied into your office or educational center, how often would you use this system? \*

Mostly  
 Prefer other biometric

Do you think if there will be accuracy issue in this system? \*

Yes  
 No  
 Maybe

Is it way good to display face recognition accuracy on screen? \*

Yes  
 No  
 Not Sure

Do you think face recognition attendance system can overcome other biometric system? \*

Yes  
 No  
 Maybe

Do you think Face Recognition Attendance system has been used by any organization/institute yet? \*

Yes  
 No  
 Maybe

Figure 151: Post-Survey Form (ii)

Do you recommend to use this system to others? \*

Yes  
 No  
 May be

In your opinion, how feasible is face recognition attendance system in context of Nepal? \*

High  
 Medium  
 Low

Is it favorable in the present condition of Coronavirus Pandemic? \*

Yes  
 No  
 Maybe

Do you think this project eliminate problem related to traditional attendance system(paper-pen attendance)? \*

Yes  
 No  
 Maybe

Please, do rate this system \*

1    2    3    4    5    6    7    8    9    10

Figure 152: Post-Survey Form (iii)

What feedback or suggestion would you prefer to provide?

Your answer

Submit      Clear form

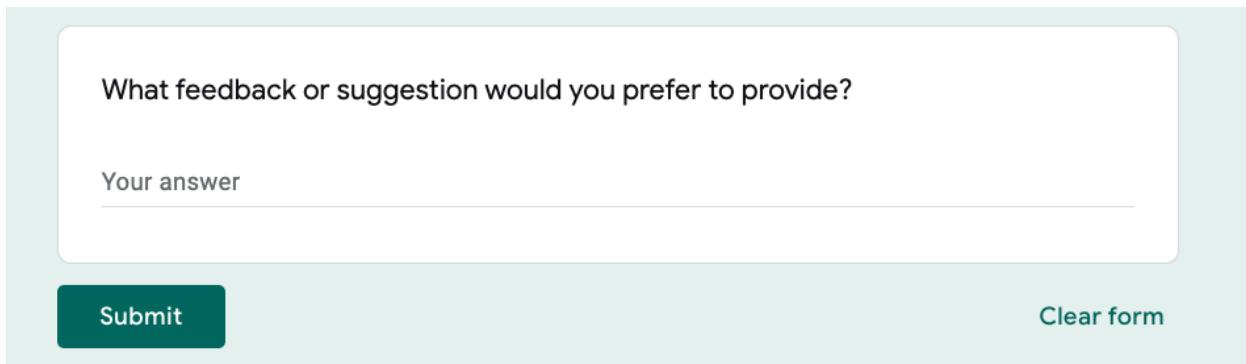


Figure 153: Post-Survey Form (iv)

### 8.5.2 Sample Of Filled Post-Survey Forms

12 responses

Accepting responses

Summary      Question      Individual

Who has responded?

Email
creativeinfotechsolution683@gmail.com
santos.dc101@gmail.com
be.hype069@gmail.com
santos.dc069@gmail.com
ybinod857@gmail.com
mahatosapana56@gamil.com
oneplusaakash9r@gmail.com
sapkotaalisha43@gmail.com
anubhavkudua20@gmail.com

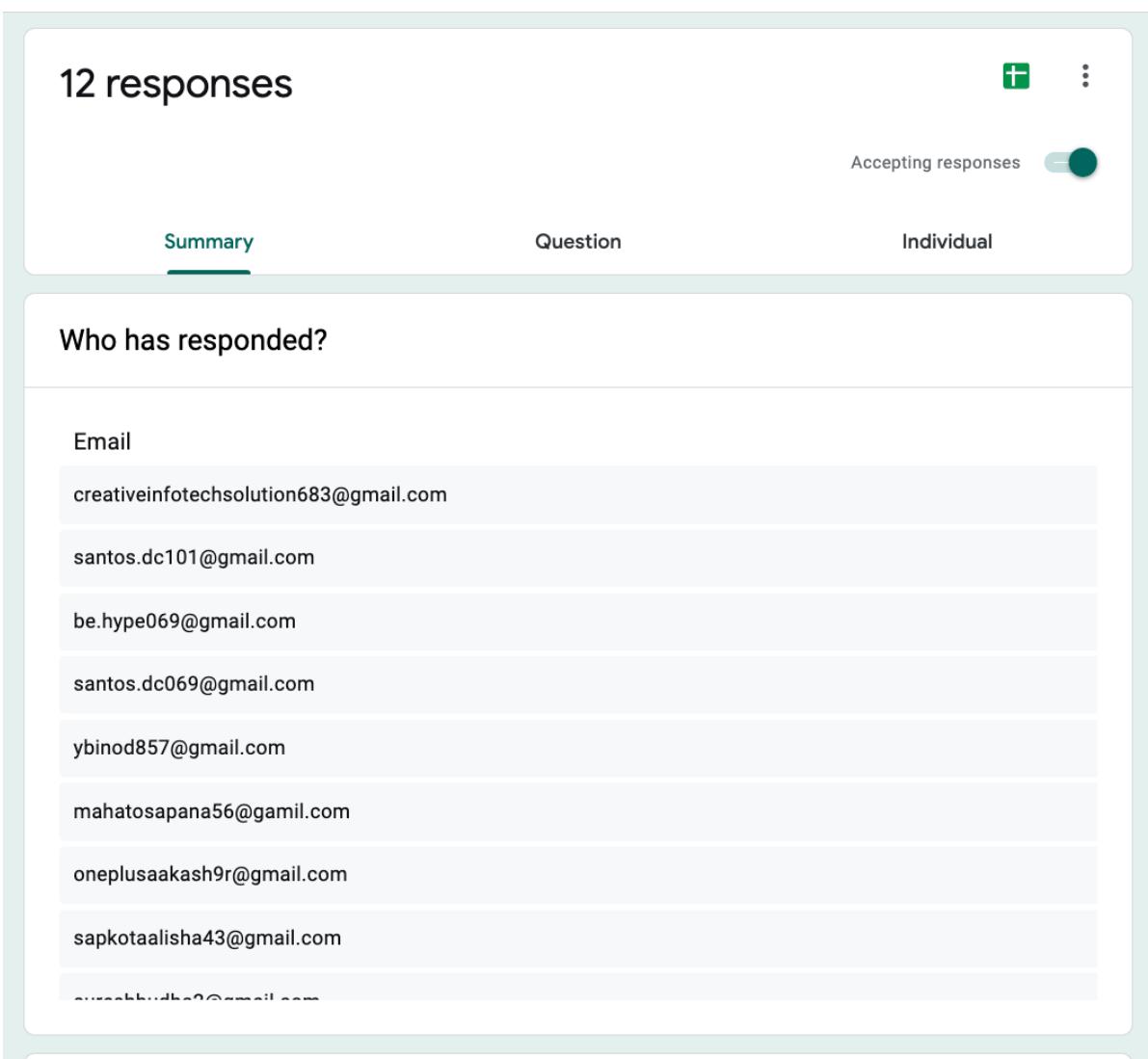


Figure 154: Post-Survey Responses (i)

sureshbudha2@gmail.com  
premoli520@gmail.com  
seraph446@gmail.com  
karanchy227@gmail.com

Figure 155: Post-Survey Responses (ii)

Name  
12 responses

Creative Infotech Solution Pvt. Ltd.  
Santosh Chaudhary  
Prateek Khadka  
Santosh DC  
Binod Yadav  
Sapana Mahato  
Khulasky  
Alisha  
Suresh Budha

Figure 156: Post-Survey Responses (iii)

Prem Oli  
Seraph Shrestha  
Karan Chaudhary

Figure 157: Post-Survey Responses (iv)

Have you ever used Face Recognition Attendance system?

12 responses

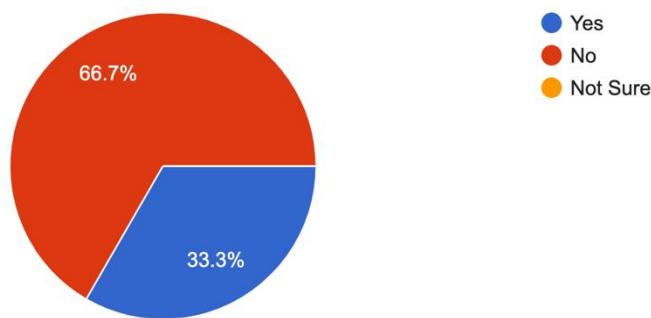


Figure 158: Post-Survey Responses (v)

If it is applied into your office or educational center, how often would you use this system?

12 responses

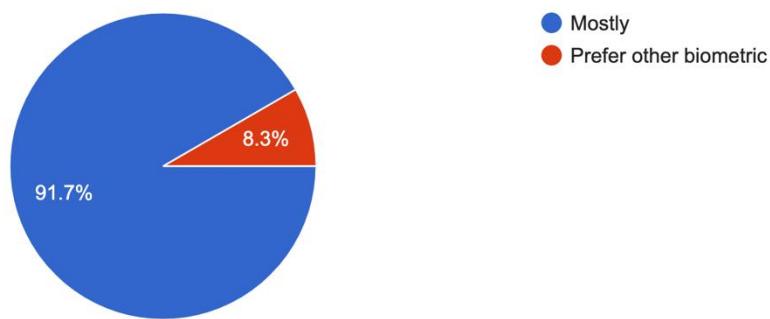


Figure 159: Post-Survey Responses (vi)

Do you think if there will be accuracy issue in this system?

12 responses

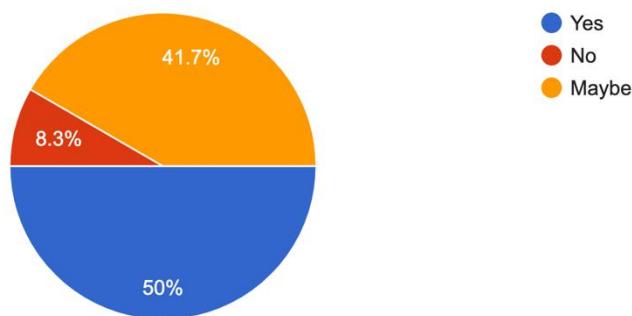


Figure 160: Post-Survey Responses (vii)

Is it way good to display face recognition accuracy on screen?

12 responses

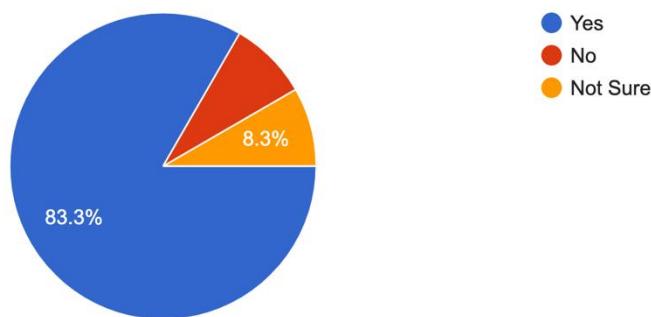


Figure 161: Post-Survey Responses (viii)

Do you think face recognition attendance system can overcome other biometric system?  
12 responses

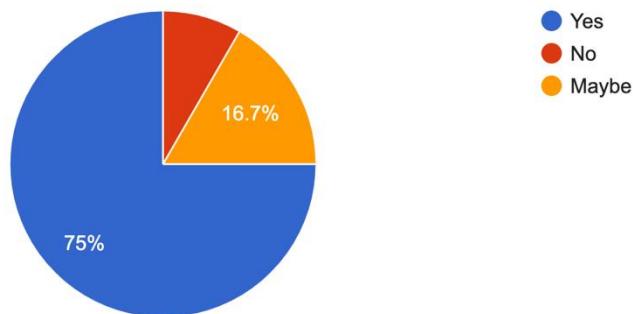


Figure 162: Post-Survey Responses (ix)

Do you think Face Recognition Attendance system has been used by any organization/institute yet?  
12 responses

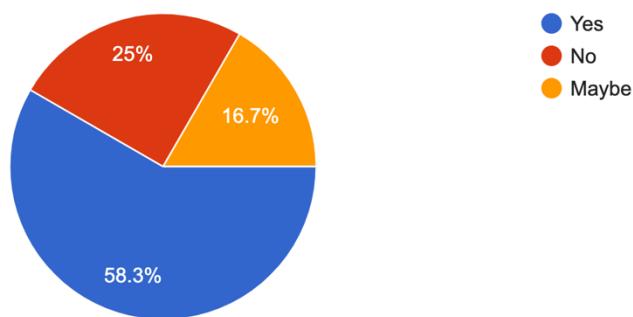


Figure 163: Post-Survey Responses (x)

Do you recommend to use this system to others?

12 responses

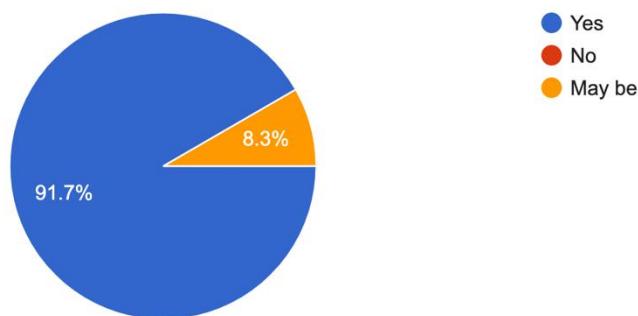


Figure 164: Post-Survey Responses (xi)

In your opinion, how feasible is face recognition attendance system in context of Nepal?

12 responses

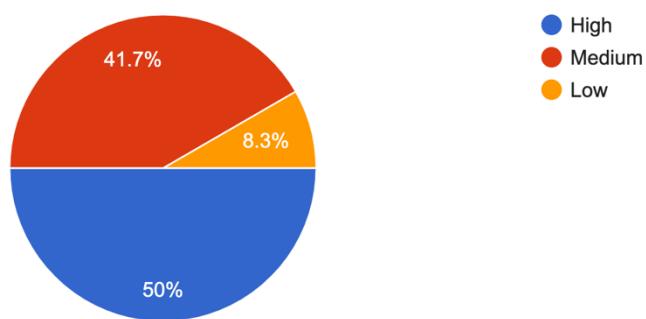


Figure 165: Post-Survey Responses (xii)

Is it favorable in the present condition of Coronavirus Pandemic?

12 responses

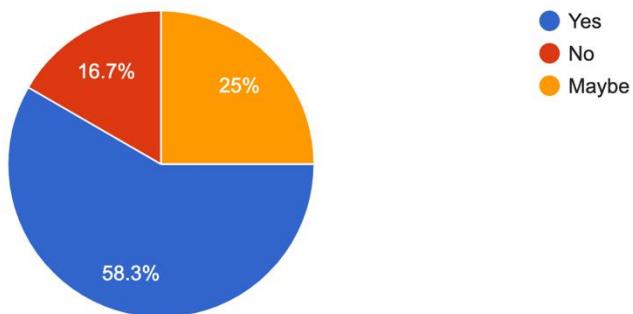


Figure 166: Post-Survey Responses (xiii)

Do you think this project eliminate problem related to traditional attendance system(paper-pen attendance)?

12 responses

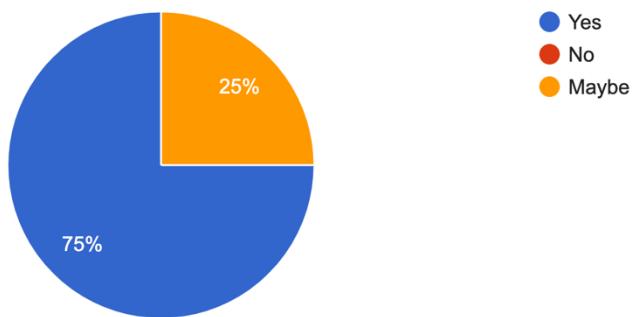


Figure 167: Post-Survey Responses (xiv)

Please, do rate this system

12 responses

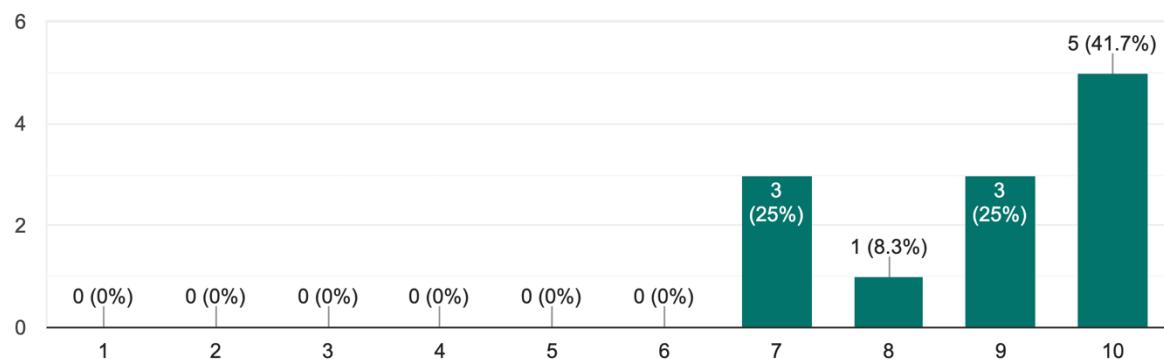


Figure 168: Post-Survey Responses (xv)

What feedback or suggestion would you prefer to provide?

4 responses

Great project comes with great demand. Thanks Mr. Dharma Raj Chaudhary  
Proud to be your client.

Great project, can't wait to try this system

Modern problem need modern solution, great job

Nope

Figure 169: Post-Survey Responses (xvi)

### 8.5.3 Post-Survey Result

Afterward the both surveys, the following is the analysis which has been driven.

- Among all, we've just received 12 responses including client too.
- Among all, 66.7% has used this project for attendance.
- 91.7% responders prefer to use this system into their office or institute in daily basis.
- As this system might get fault in accuracy somehow, so 50% responders questioned accuracy issue.
- Most of the responder liked the system to display face matching accuracy percentage on screen.
- 75% of people thinks that this system might overcome other biometric systems.
- Majority thinks that this type of system has been using in companies and organization.
- Most of them liked to recommend it to others.
- It is somehow feasible in context of Nepal.
- It might be favorable in context of pandemic though student can remove mask for a few second.
- Among all, 41% rated it 10 out of 10 whereas 25% rated 25%.
- There are some feedbacks which has been attached above.

## 8.6 Appendix C: Sample Codes

### 8.6.1 Sample Code of the UI

#### 8.6.1.1 Login Window

```

login.py
1  import email
2  from tkinter import Tk, font, messagebox
3  from tkinter import ttk
4  from time import strftime
5  from PIL import Image,ImageTk
6  import mysql.connector
7  from Home import home_window
8
9
10 def main():
11     win=Tk()
12     app=Login_Window(win)
13     win.mainloop()
14
15 class Login_Window:
16     def __init__(self,root):
17         self.root=root
18         self.root.title("ERAS")
19         self.root.geometry("1200x650+0+0")
20
21     #BG image
22         self.bg=ImageTk.PhotoImage(File="/home/hype/Downloads/FYP-Development/VS-Code/Images/login_bg.jpg")
23
24         lbl_bg=Label(self.root,image=self.bg)
25         lbl_bg.place(x=0,y=0,width=1200,height=650)
26
27     #Frame
28         login_frame=Frame(self.root,bg="black")
29         login_frame.place(x=420,y=70,width=340,height=450)
30

```

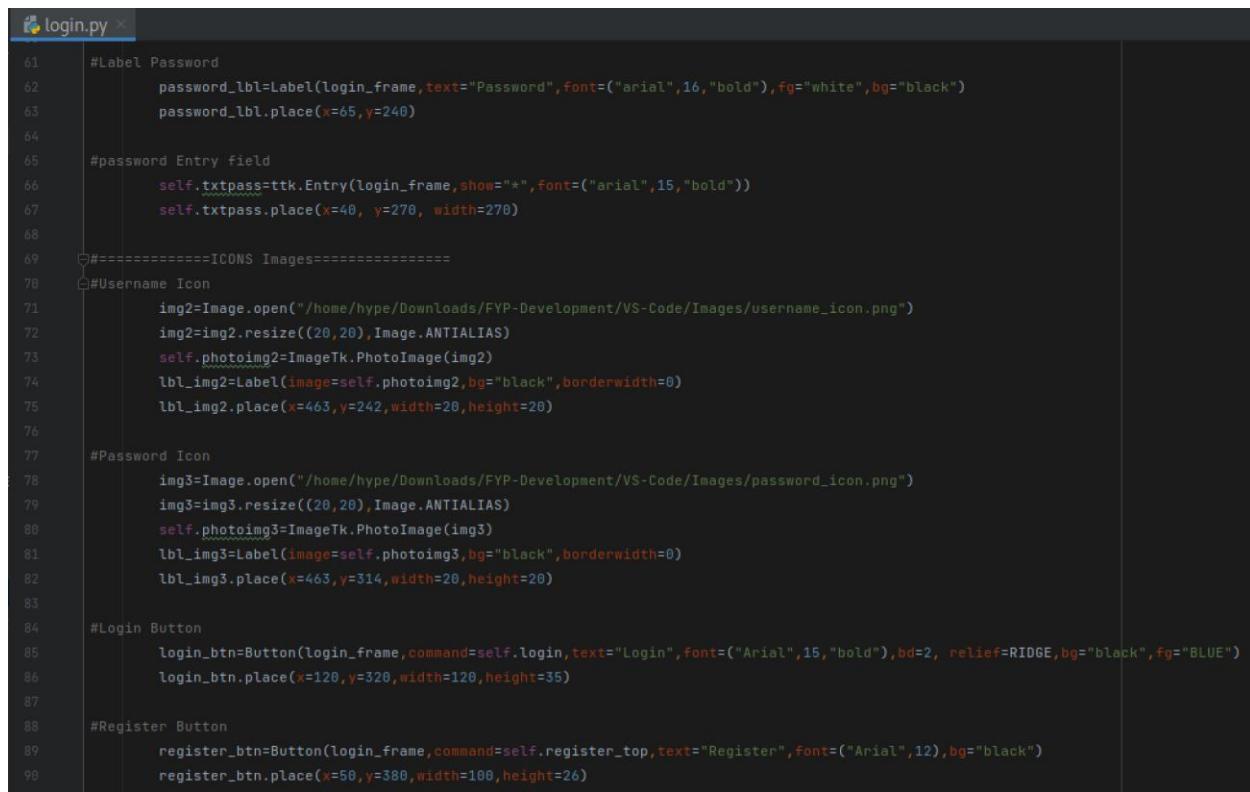
Figure 170: Login Window (i)

```

login.py
31     #Time
32     def time():
33         string=strftime('%H:%M:%S %p')
34         lbl.config(text=string)
35         lbl.after(1000, time)
36
37         lbl=Label(lbl_bg,font=_,("arial",15,"bold"),bg="black",fg="white")
38         lbl.place(x=418,y=44,width=130,height=25)
39         time()
40
41
42     #Inside Frame
43         img1=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/login_icon.png")
44         img1=img1.resize((100,100),Image.ANTIALIAS)
45         self.photoimg1=ImageTk.PhotoImage(img1)
46         lbl_img1=Label(image=self.photoimg1,bg="black",borderwidth=0)
47         lbl_img1.place(x=550,y=90,width=100,height=100)
48
49     #Label Below Icon
50         get_str=Label(login_frame,text="Get Started",font=(_("arial",20,"bold"),foreground="white",bg="black"))
51         get_str.place(x=120,y=120)
52
53     #Label Username
54         username_lbl=Label(login_frame,text="Email",font=(_("arial",16,"bold"),fg="white",bg="black"))
55         username_lbl.place(x=65,y=170)
56
57     #username Entry field
58         self.txtuser=ttk.Entry(login_frame,font=(_("arial",15,"bold")))
59         self.txtuser.place(x=40, y=200, width=270)
60

```

Figure 171: Login Window (ii)

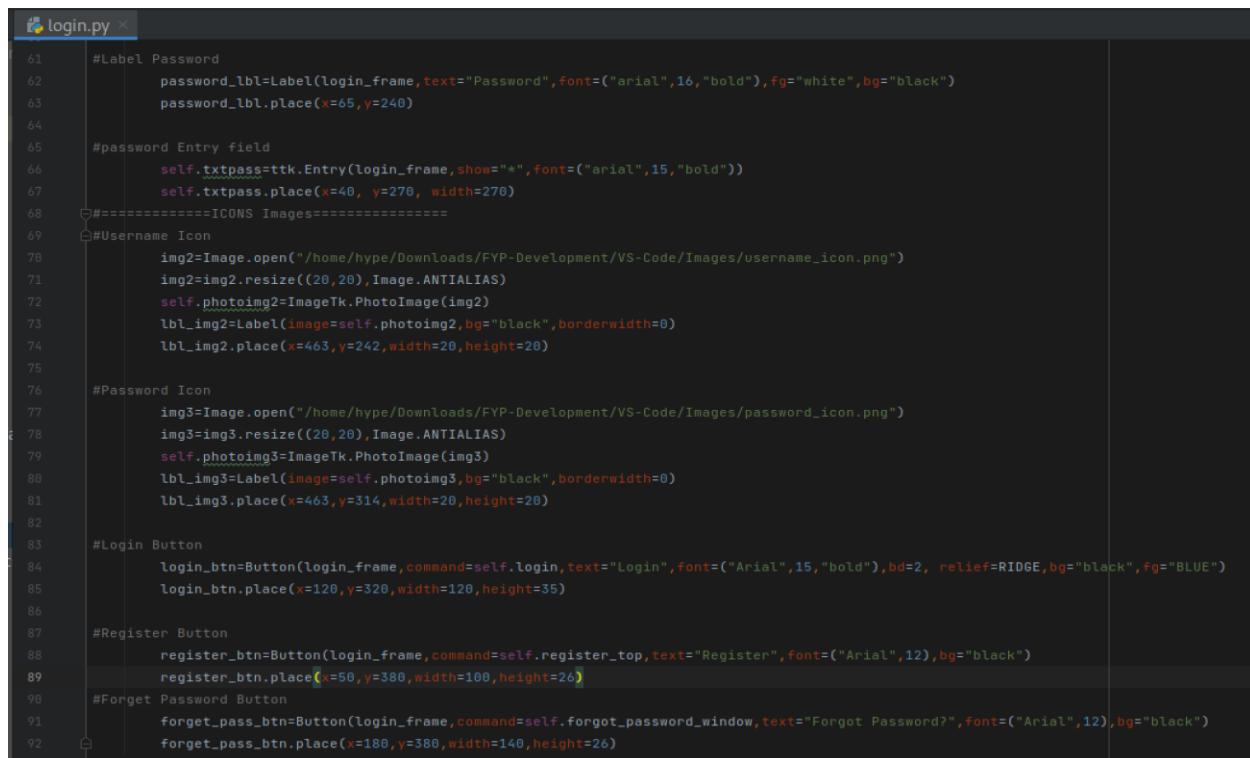


```

 61     #Label Password
 62     password_lbl=Label(login_frame,text="Password",font=("arial",16,"bold"),fg="white",bg="black")
 63     password_lbl.place(x=65,y=240)
 64
 65     #password Entry field
 66     self.txtpass=ttk.Entry(login_frame,show="*",font=("arial",15,"bold"))
 67     self.txtpass.place(x=40, y=270, width=270)
 68
 69     #=====ICONS Images=====
 70     #Username Icon
 71     img2=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/username_icon.png")
 72     img2=img2.resize((20,20),Image.ANTIALIAS)
 73     self.photoimg2=ImageTk.PhotoImage(img2)
 74     lbl_img2=Label(image=self.photoimg2,bg="black",borderwidth=0)
 75     lbl_img2.place(x=463,y=242,width=20,height=20)
 76
 77     #Password Icon
 78     img3=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/password_icon.png")
 79     img3=img3.resize((20,20),Image.ANTIALIAS)
 80     self.photoimg3=ImageTk.PhotoImage(img3)
 81     lbl_img3=Label(image=self.photoimg3,bg="black",borderwidth=0)
 82     lbl_img3.place(x=463,y=314,width=20,height=20)
 83
 84     #Login Button
 85     login_btn=Button(login_frame,command=self.login,text="Login",font=("Arial",15,"bold"),bd=2, relief=RIDGE,bg="black",fg="BLUE")
 86     login_btn.place(x=120,y=320,width=120,height=35)
 87
 88     #Register Button
 89     register_btn=Button(login_frame,command=self.register_top,text="Register",font=("Arial",12),bg="black")
 90     register_btn.place(x=50,y=380,width=100,height=26)

```

Figure 172: Login Window (iii)



```

 61     #Label Password
 62     password_lbl=Label(login_frame,text="Password",font=("arial",16,"bold"),fg="white",bg="black")
 63     password_lbl.place(x=65,y=240)
 64
 65     #password Entry field
 66     self.txtpass=ttk.Entry(login_frame,show="*",font=("arial",15,"bold"))
 67     self.txtpass.place(x=40, y=270, width=270)
 68     #=====ICONS Images=====
 69     #Username Icon
 70     img2=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/username_icon.png")
 71     img2=img2.resize((20,20),Image.ANTIALIAS)
 72     self.photoimg2=ImageTk.PhotoImage(img2)
 73     lbl_img2=Label(image=self.photoimg2,bg="black",borderwidth=0)
 74     lbl_img2.place(x=463,y=242,width=20,height=20)
 75
 76     #Password Icon
 77     img3=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/password_icon.png")
 78     img3=img3.resize((20,20),Image.ANTIALIAS)
 79     self.photoimg3=ImageTk.PhotoImage(img3)
 80     lbl_img3=Label(image=self.photoimg3,bg="black",borderwidth=0)
 81     lbl_img3.place(x=463,y=314,width=20,height=20)
 82
 83     #Login Button
 84     login_btn=Button(login_frame,command=self.login,text="Login",font=("Arial",15,"bold"),bd=2, relief=RIDGE,bg="black",fg="BLUE")
 85     login_btn.place(x=120,y=320,width=120,height=35)
 86
 87     #Register Button
 88     register_btn=Button(login_frame,command=self.register_top,text="Register",font=("Arial",12),bg="black")
 89     register_btn.place(x=50,y=380,width=100,height=26)
 90     #Forgot Password Button
 91     forget_pass_btn=Button(login_frame,command=self.forgot_password_window,text="Forgot Password?",font=("Arial",12),bg="black")
 92     forget_pass_btn.place(x=180,y=380,width=140,height=26)

```

Figure 173: Login Window (iv)

```
#Label
label1=Label(self.root2,text="Forgot Password",font=("Arial",20,"bold"),fg="white")
label1.place(x=0,y=10,relwidth=1)

#Security Question Combobox
security_q=Label(self.root2,text="Security Question",font=("arial",14,"bold"))
security_q.place(x=90,y=80)

self.combo_security=ttk.Combobox(self.root2,font=("arial",14),state="readonly")
self.combo_security["values"]=[("Select","Your birth place?","Your pet name?","Your favorite food?")]
self.combo_security.place(x=90,y=110,width=180)
self.combo_security.current(0)

#Security Answer
security_ans=Label(self.root2,text="Security Answer",font=("arial",14,"bold"))
security_ans.place(x=90,y=150)

self.security_ans_entry=ttk.Entry(self.root2,font=("arial",13,"bold"))
self.security_ans_entry.place(x=90,y=180,width=200)

#New Password
new_password=Label(self.root2,text="New Password",font=("arial",14,"bold"))
new_password.place(x=90,y=220)

self.new_password_entry=ttk.Entry(self.root2,font=("arial",13,"bold"))
self.new_password_entry.place(x=90,y=250,width=200)

#Save Button
save_button=Button(self.root2,command=self.reset_pass,text="Reset",font=("Arial",15,"bold"),fg="white",bg="black")
save_button.place(x=150,y=300)
```

Figure 174: Login Window (v)

### 8.6.1.2 Register window

```

===== R E G I S T E R   W I N D O W   C L A S S =====
class Register_window:
    def __init__(self,root):
        self.root = root
        self.root.title("Register")
        self.root.geometry("1200x650+0+0")

    #Text Variable
        self.var_first_name=StringVar()
        self.var_last_name=StringVar()
        self.var_contact=StringVar()
        self.var_email=StringVar()
        self.var_pass=StringVar()
        self.var_confirm_pass=StringVar()
        self.var_security_qst=StringVar()
        self.var_security_ans=StringVar()
        self.var_check_button=IntVar()

    #BG
        self.bg=ImageTk.PhotoImage(file="/home/hype/Downloads/FYP-Development/VS-Code/Images/register_bg.jpg")
        bg_lbl=Label(self.root,image=self.bg)
        bg_lbl.place(x=0,y=0,relwidth=1,relheight=1)

    #Left Image
        self.bg1=ImageTk.PhotoImage(file="/home/hype/Downloads/FYP-Development/VS-Code/Images/register_left.jpg")
        left_lbl=Label(self.root,image=self.bg1, bg="black")
        left_lbl.place(x=100,y=50,width=500,height=520)

    #Right Frame
        right_frame=Frame(self.root, bg="black")
        right_frame.place(x=600,y=50,width=500,height=520)

```

Figure 175: Register window (i)

```
236     #Label Register
237     register_lbl=Label(right_frame,text="Register Now",font=("Arial",28,"bold"),bg="black",fg="red")
238     register_lbl.place(x=20,y=20)
239
240     #Label and Entry Fields
241     #First Name-----
242     first_name=Label(right_frame,text="First Name",font=("arial",14),bg="black")
243     first_name.place(x=20,y=70)
244
245     first_Name_entry=ttk.Entry(right_frame,textvariable=self.var_first_name,font=("arial",13))
246     first_Name_entry.place(x=20,y=95,width=200)
247
248     #Last Name
249     last_name=Label(right_frame,text="Last Name",font=("arial",14),bg="black")
250     last_name.place(x=250,y=70)
251
252     last_Name_entry=ttk.Entry(right_frame,textvariable=self.var_last_name,font=("arial",13))
253     last_Name_entry.place(x=250,y=95,width=200)
254
255     #Contact No.
256     contact=Label(right_frame,text="Contact No.",font=("arial",14),bg="black")
257     contact.place(x=20,y=140)
258
259     contact_entry=ttk.Entry(right_frame,textvariable=self.var_contact,font=("arial",13))
260     contact_entry.place(x=20,y=165,width=200)
261
262     #Email
263     email=Label(right_frame,text="Email",font=("arial",14),bg="black")
264     email.place(x=250,y=140)
265
266     email_entry=ttk.Entry(right_frame,textvariable=self.var_email,font=("arial",13))
267     email_entry.place(x=250,y=165,width=200)
```

Figure 176: Register window (ii)

```
#Password
password=Label(right_frame,text="Password",font=("arial",14),bg="black")
password.place(x=20,y=210)

password_entry=ttk.Entry(right_frame,show="*",textvariable=self.var_pass,font=("arial",13))
password_entry.place(x=20,y=235,width=200)

#Confirm Password
password=Label(right_frame,text="Confirm Password",font=("arial",14),bg="black")
password.place(x=250,y=210)

password_entry=ttk.Entry(right_frame,show="*",textvariable=self.var_confirm_pass,font=("arial",13))
password_entry.place(x=250,y=235,width=200)

#Security Question Combobox
security_qstn=Label(right_frame,text="Security Question",font=("arial",14),bg="black")
security_qstn.place(x=20,y=280)

combo_security=ttk.Combobox(right_frame,textvariable=self.var_security_qst,font=("arial",14),state="readonly")
combo_security["values"]=("Select","Your birth place?","Your pet name?","Your favorite food?")
combo_security.place(x=20,y=305,width=180)
combo_security.current(0)

#Security Answer
security_ans=Label(right_frame,text="Security Answer",font=("arial",14),bg="black")
security_ans.place(x=250,y=280)

security_ans_entry=ttk.Entry(right_frame,textvariable=self.var_security_ans,font=("arial",13))
security_ans_entry.place(x=250,y=305,width=200)
```

Figure 177: Register window (iii)

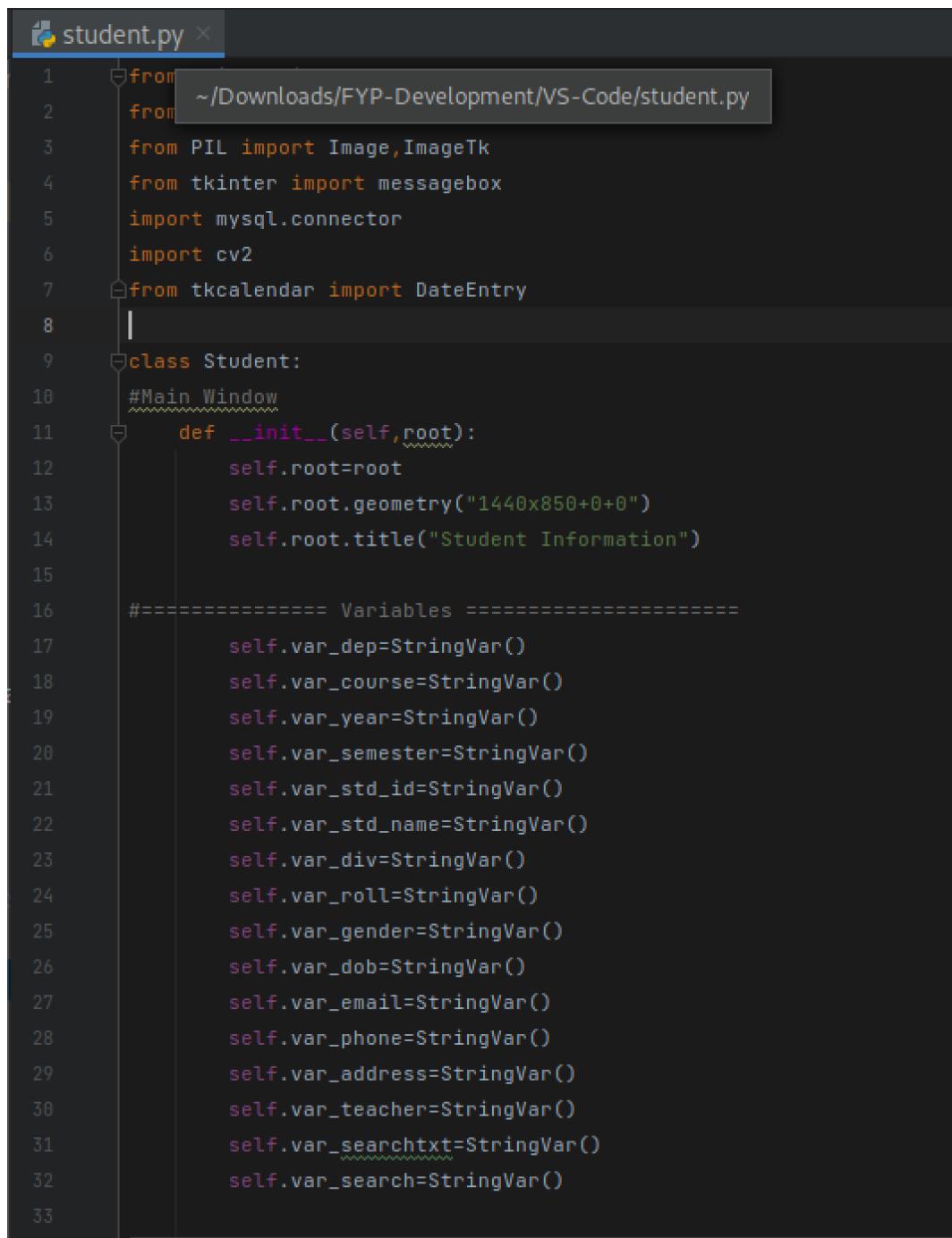
```
#Check Button
check_button=Checkbutton(right_frame,variable=self.var_check_button,text="I agree all the terms & conditions.",font=("Arial",10),bg="black",onvalue=1,
check_button.place(x=20,y=350)

#Register Button
img=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/register_button.png")
img=img.resize((150,30),Image.ANTIALIAS)
self.photoimage=ImageTk.PhotoImage(img)

reg_button=Button(right_frame,command=self.register_data,image=self.photoimage,borderwidth=0,cursor="hand2")
reg_button.place(x=20,y=390,width=150)
```

Figure 178: Register window (iv)

### 8.6.1.3 Student Window



A screenshot of a Visual Studio Code editor window titled "student.py". The code is written in Python and defines a class "Student" for a main window. The code includes imports for PIL, tkinter, mysql.connector, cv2, and tkcalendar. It sets up variables for student information like department, course, year, semester, ID, name, division, roll number, gender, dob, email, phone, address, teacher, search text, and search. The code uses StringVar() for all these variables.

```
1  from ~ / Downloads / FYP - Development / VS - Code / student . py
2  from PIL import Image, ImageTk
3  from tkinter import messagebox
4  import mysql.connector
5  import cv2
6  from tkcalendar import DateEntry
7
8
9  class Student:
10     # Main Window
11     def __init__(self, root):
12         self.root = root
13         self.root.geometry("1440x850+0+0")
14         self.root.title("Student Information")
15
16     # ===== Variables =====
17     self.var_dep = StringVar()
18     self.var_course = StringVar()
19     self.var_year = StringVar()
20     self.var_semester = StringVar()
21     self.var_std_id = StringVar()
22     self.var_std_name = StringVar()
23     self.var_div = StringVar()
24     self.var_roll = StringVar()
25     self.var_gender = StringVar()
26     self.var_dob = StringVar()
27     self.var_email = StringVar()
28     self.var_phone = StringVar()
29     self.var_address = StringVar()
30     self.var_teacher = StringVar()
31     self.var_searchtxt = StringVar()
32     self.var_search = StringVar()
33
```

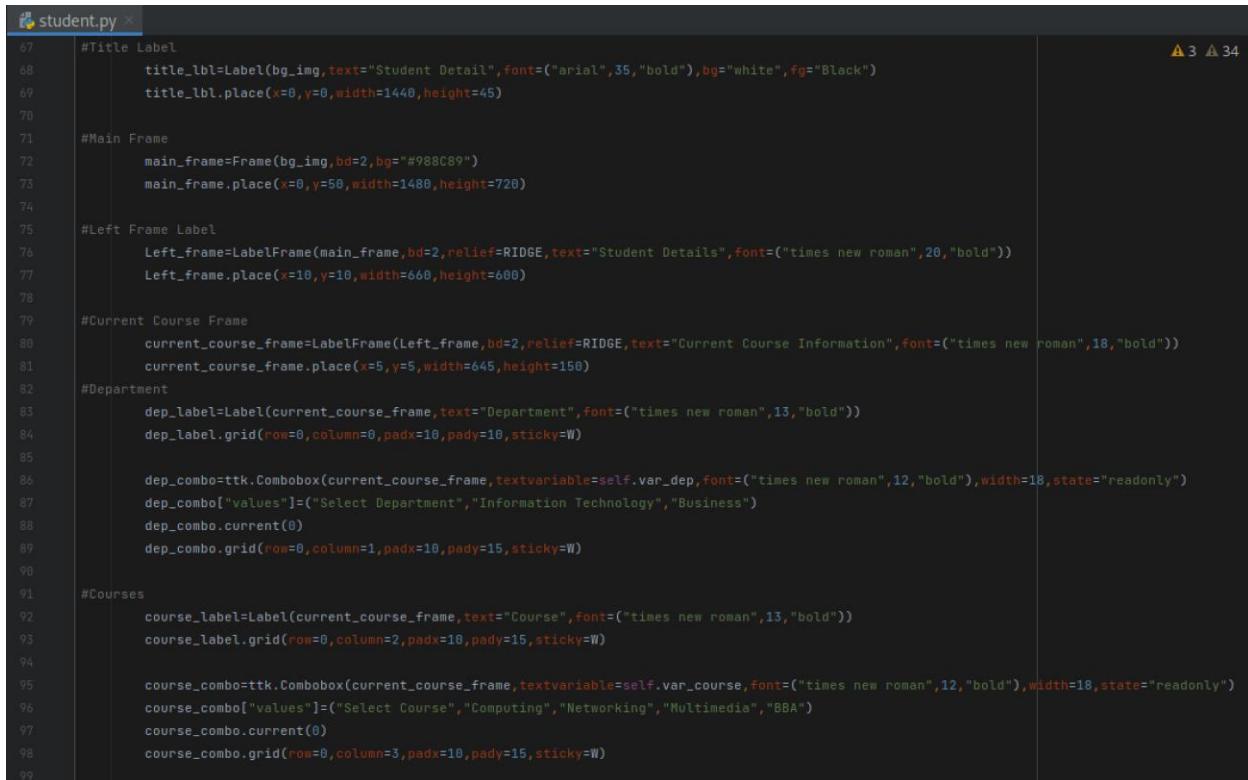
Figure 179: Register window (v)

```

student.py ×
34     #Cover Image1
35         cover1=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover1.png")
36         #cover1=Image.open("Images/cover1.png")
37         cover1=cover1.resize((480,130),Image.ANTIALIAS)
38         self.coverimg1=ImageTk.PhotoImage(cover1)
39
40         cover1_lbl=Label(self.root,image=self.coverimg1)
41         cover1_lbl.place(x=0,y=0,width=480, height=130)
42
43     #Cover Image 2
44         cover2=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover2.png")
45         cover2=cover2.resize((480,130),Image.ANTIALIAS)
46         self.cover_img2=ImageTk.PhotoImage(cover2)
47
48         cover2_lbl=Label(self.root,image=self.cover_img2)
49         cover2_lbl.place(x=480, y=0,width=480, height=130)
50
51     #Cover Image 3
52         cover3=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover3.jpeg")
53         cover3=cover3.resize((480,130),Image.ANTIALIAS)
54         self.coverimg3 = ImageTk.PhotoImage(cover3)
55
56         cover3_lbl=Label(self.root,image=self.coverimg3)
57         cover3_lbl.place(x=960,y=0,width=480, height=130)
58
59     #Main Image 3
60         main=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/main.jpg")
61         main=main.resize((1440,720),Image.ANTIALIAS)
62         self.main_img=ImageTk.PhotoImage(main)
63
64         bg_img=Label(self.root,image=self.main_img)
65         bg_img.place(x=0,y=150,width=1440, height=720)
66

```

Figure 180: Register window (vi)

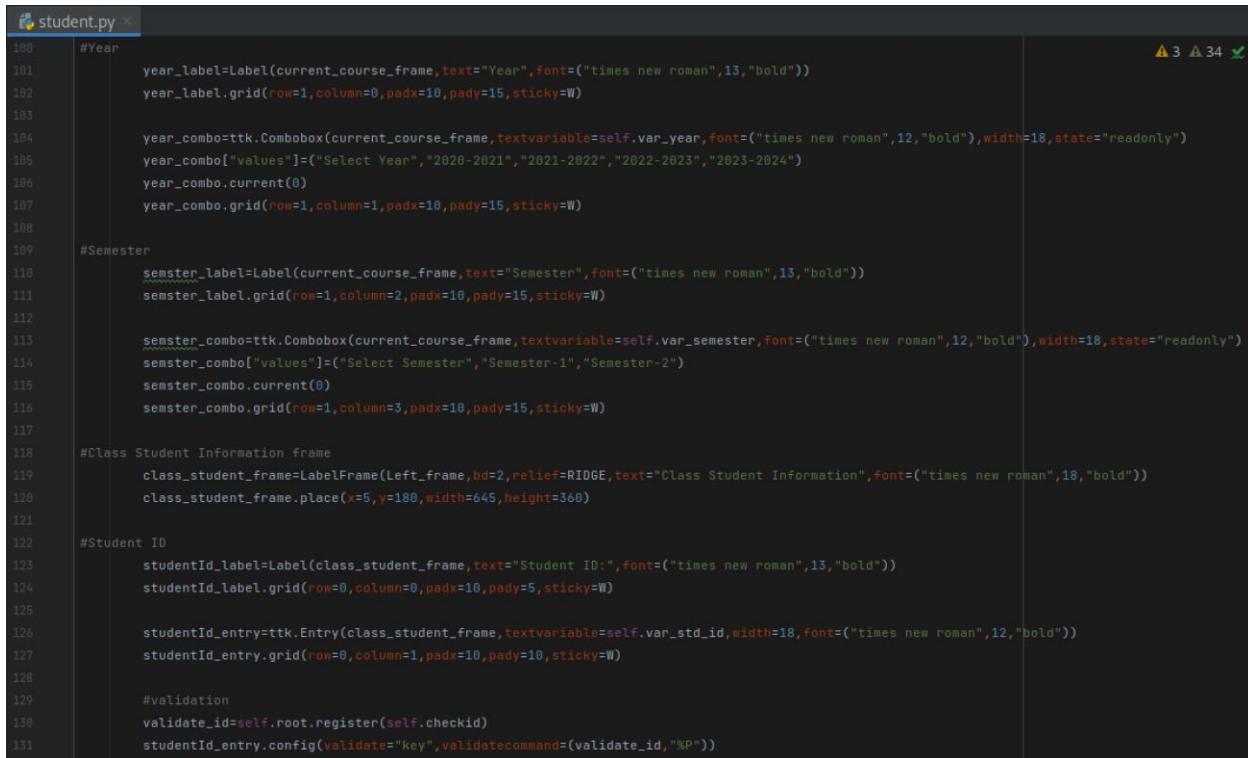


```

67     #Title Label
68         title_lbl=Label(bg_img,text="Student Detail",font=("arial",35,"bold"),bg="white",fg="Black")
69         title_lbl.place(x=0,y=0,width=1440,height=45)
70
71     #Main Frame
72         main_frame=Frame(bg_img,bd=2,bg="#988C89")
73         main_frame.place(x=0,y=50,width=1480,height=720)
74
75     #Left Frame Label
76         Left_frame=LabelFrame(main_frame,bd=2,relief=RIDGE,font="times new roman",20,"bold"))
77         Left_frame.place(x=10,y=10,width=660,height=600)
78
79     #Current Course Frame
80         current_course_frame=LabelFrame(Left_frame,bd=2,relief=RIDGE,font="times new roman",18,"bold"))
81         current_course_frame.place(x=5,y=5,width=645,height=150)
82
83     #Department
84         dep_label=Label(current_course_frame,text="Department",font="times new roman",13,"bold"))
85         dep_label.grid(row=0,column=0,padx=10,pady=10,sticky=W)
86
87         dep_combo=ttk.Combobox(current_course_frame,textvariable=self.var_dep,font="times new roman",12,"bold"),width=18,state="readonly")
88         dep_combo["values"]=( "Select Department", "Information Technology", "Business")
89         dep_combo.current(0)
90         dep_combo.grid(row=0,column=1,padx=10,pady=15,sticky=W)
91
92     #Courses
93         course_label=Label(current_course_frame,text="Course",font="times new roman",13,"bold"))
94         course_label.grid(row=0,column=2,padx=10,pady=15,sticky=W)
95
96         course_combo=ttk.Combobox(current_course_frame,textvariable=self.var_course,font="times new roman",12,"bold"),width=18,state="readonly")
97         course_combo["values"]=( "Select Course", "Computing", "Networking", "Multimedia", "BBA")
98         course_combo.current(0)
99         course_combo.grid(row=0,column=3,padx=10,pady=15,sticky=W)

```

Figure 181: Register window (vii)

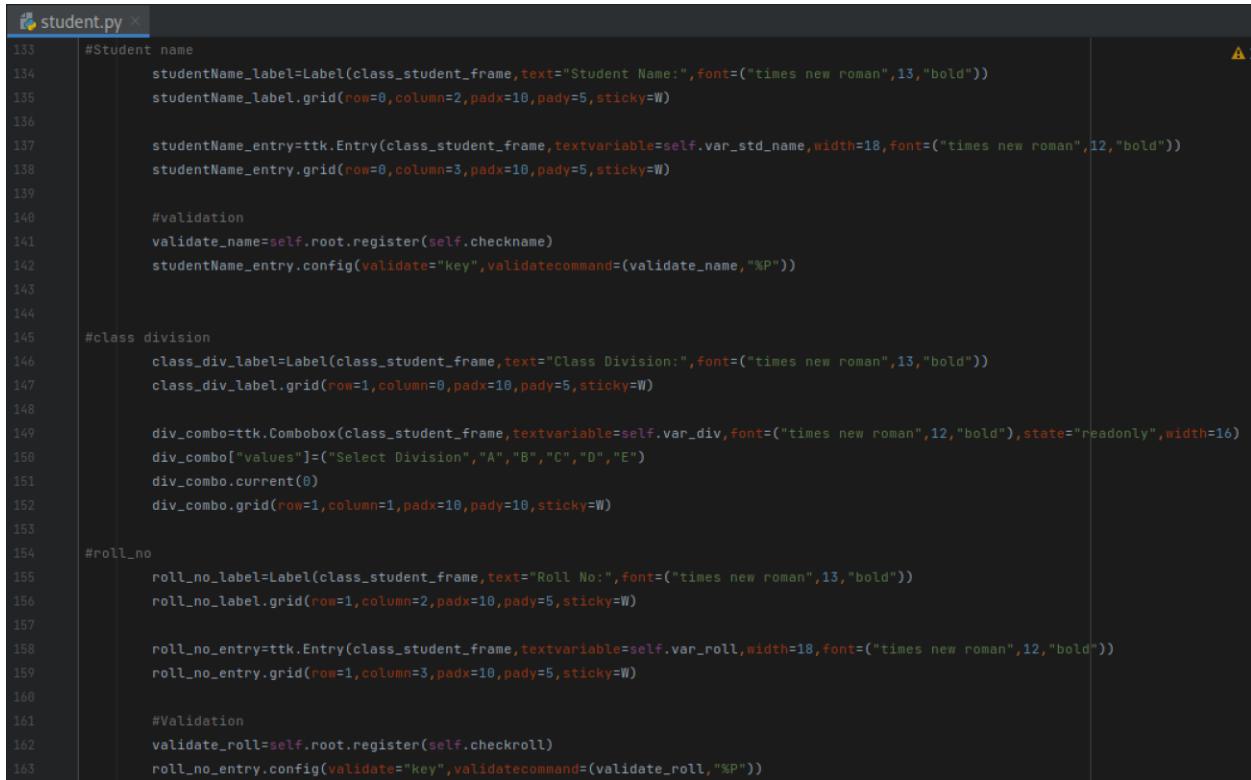


```

100    #Year
101        year_label=Label(current_course_frame,text="Year",font="times new roman",13,"bold"))
102        year_label.grid(row=1,column=0,padx=10,pady=15,sticky=W)
103
104        year_combo=ttk.Combobox(current_course_frame,textvariable=self.var_year,font="times new roman",12,"bold"),width=18,state="readonly")
105        year_combo["values"]=( "Select Year", "2020-2021", "2021-2022", "2022-2023", "2023-2024")
106        year_combo.current(0)
107        year_combo.grid(row=1,column=1,padx=10,pady=15,sticky=W)
108
109    #Semester
110        semster_label=Label(current_course_frame,text="Semester",font="times new roman",13,"bold"))
111        semster_label.grid(row=1,column=2,padx=10,pady=15,sticky=W)
112
113        semster_combo=ttk.Combobox(current_course_frame,textvariable=self.var_semester,font="times new roman",12,"bold"),width=18,state="readonly")
114        semster_combo["values"]=( "Select Semester", "Semester-1", "Semester-2")
115        semster_combo.current(0)
116        semster_combo.grid(row=1,column=3,padx=10,pady=15,sticky=W)
117
118    #Class Student Information frame
119        class_student_frame=LabelFrame(Left_frame,bd=2,relief=RIDGE,font="times new roman",18,"bold"))
120        class_student_frame.place(x=5,y=180,width=645,height=360)
121
122    #Student ID
123        studentId_label=Label(class_student_frame,text="Student ID:",font="times new roman",13,"bold"))
124        studentId_label.grid(row=0,column=0,padx=10,pady=5,sticky=W)
125
126        studentId_entry=ttk.Entry(class_student_frame,textvariable=self.var_std_id,font="times new roman",12,"bold"))
127        studentId_entry.grid(row=0,column=1,padx=10,pady=10,sticky=W)
128
129        #validation
130        validate_id=self.root.register(self.checkid)
131        studentId_entry.config(validate="key",validatecommand=(validate_id,"%P"))

```

Figure 182: Register window (viii)

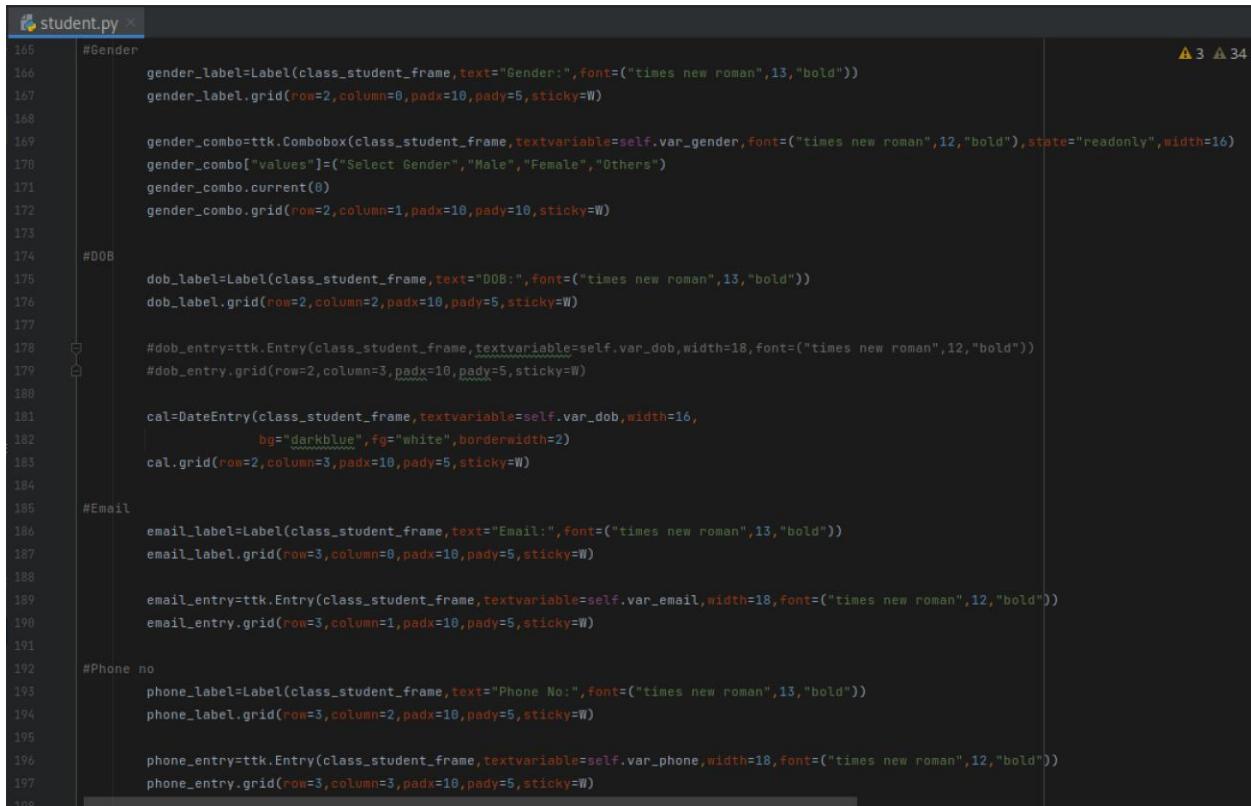


```

133     #Student name
134     studentName_label=Label(class_student_frame,text="Student Name:",font=("times new roman",13,"bold"))
135     studentName_label.grid(row=0,column=2,padx=10,pady=5,sticky=W)
136
137     studentName_entry=ttk.Entry(class_student_frame,textvariable=self.var_std_name,width=18,font=("times new roman",12,"bold"))
138     studentName_entry.grid(row=0,column=3,padx=10,pady=5,sticky=W)
139
140     #validation
141     validate_name=self.root.register(self.checkname)
142     studentName_entry.config(validate="key",validatecommand=(validate_name,"%P"))
143
144
145     #class division
146     class_div_label=Label(class_student_frame,text="Class Division:",font=("times new roman",13,"bold"))
147     class_div_label.grid(row=1,column=0,padx=10,pady=5,sticky=W)
148
149     div_combo=ttk.Combobox(class_student_frame,textvariable=self.var_div,font=("times new roman",12,"bold"),state="readonly",width=16)
150     div_combo["values"]=("Select Division","A","B","C","D","E")
151     div_combo.current(0)
152     div_combo.grid(row=1,column=1,padx=10,pady=10,sticky=W)
153
154     #roll_no
155     roll_no_label=Label(class_student_frame,text="Roll No:",font=("times new roman",13,"bold"))
156     roll_no_label.grid(row=1,column=2,padx=10,pady=5,sticky=W)
157
158     roll_no_entry=ttk.Entry(class_student_frame,textvariable=self.var_roll,width=18,font=("times new roman",12,"bold"))
159     roll_no_entry.grid(row=1,column=3,padx=10,pady=5,sticky=W)
160
161     #Validation
162     validate_roll=self.root.register(self.checkroll)
163     roll_no_entry.config(validate="key",validatecommand=(validate_roll,"%P"))

```

Figure 183: Register window (ix)



```

165     #Gender
166     gender_label=Label(class_student_frame,text="Gender:",font=("times new roman",13,"bold"))
167     gender_label.grid(row=2,column=0,padx=10,pady=5,sticky=W)
168
169     gender_combo=ttk.Combobox(class_student_frame,textvariable=self.var_gender,font=("times new roman",12,"bold"),state="readonly",width=16)
170     gender_combo["values"]=("Select Gender","Male","Female","Others")
171     gender_combo.current(0)
172     gender_combo.grid(row=2,column=1,padx=10,pady=10,sticky=W)
173
174     #DOB
175     dob_label=Label(class_student_frame,text="DOB:",font=("times new roman",13,"bold"))
176     dob_label.grid(row=2,column=2,padx=10,pady=5,sticky=W)
177
178     #dob_entry=ttk.Entry(class_student_frame,textvariable=self.var_dob,width=18,font=("times new roman",12,"bold"))
179     #dob_entry.grid(row=2,column=3,padx=10,pady=5,sticky=W)
180
181     cal=DateEntry(class_student_frame,textvariable=self.var_dob,width=16,
182                   bg="darkblue",fg="white",borderwidth=2)
183     cal.grid(row=2,column=3,padx=10,pady=5,sticky=W)
184
185     #Email
186     email_label=Label(class_student_frame,text="Email:",font=("times new roman",13,"bold"))
187     email_label.grid(row=3,column=0,padx=10,pady=5,sticky=W)
188
189     email_entry=ttk.Entry(class_student_frame,textvariable=self.var_email,width=18,font=("times new roman",12,"bold"))
190     email_entry.grid(row=3,column=1,padx=10,pady=5,sticky=W)
191
192     #Phone no
193     phone_label=Label(class_student_frame,text="Phone No:",font=("times new roman",13,"bold"))
194     phone_label.grid(row=3,column=2,padx=10,pady=5,sticky=W)
195
196     phone_entry=ttk.Entry(class_student_frame,textvariable=self.var_phone,width=18,font=("times new roman",12,"bold"))
197     phone_entry.grid(row=3,column=3,padx=10,pady=5,sticky=W)

```

Figure 184: Register window (x)

```

198     phone_entry.grid(row=4, column=0, padx=10, pady=5, sticky=W)
199
200     #Validation
201     validate_phone=self.root.register(self.checkphone)
202     phone_entry.config(validate="key", validatecommand=(validate_phone,"%P"))
203
204     #Address
205     address_label=Label(class_student_frame,text="Address:",font=("times new roman",13,"bold"))
206     address_label.grid(row=4,column=0,padx=10,pady=5,sticky=W)
207
208     address_entry=ttk.Entry(class_student_frame,textvariable=self.var_address,width=18,font=("times new roman",12,"bold"))
209     address_entry.grid(row=4,column=1,padx=10,pady=5,sticky=W)
210
211     #validation
212     validate_address=self.root.register(self.checkaddress)
213     address_entry.config(validate="key", validatecommand=(validate_address,"%P"))
214
215     #Teacher Name
216     teacher_label=Label(class_student_frame,text="Teacher Name:",font=("times new roman",13,"bold"))
217     teacher_label.grid(row=4,column=2,padx=10,pady=5,sticky=W)
218
219     teacher_entry=ttk.Entry(class_student_frame,textvariable=self.var_teacher,width=18,font=("times new roman",12,"bold"))
220     teacher_entry.grid(row=4,column=3,padx=10,pady=5,sticky=W)
221
222     #validation
223     validate_teacher=self.root.register(self.checkteachername)
224     teacher_entry.config(validate="key", validatecommand=(validate_teacher,"%P"))
225
226     #radio buttons
227     self.var_radio1=StringVar()
228     radiobtn1=ttk.Radiobutton(class_student_frame,variable=self.var_radio1,text="Take Photo Sample",value="Yes")
229     radiobtn1.grid(row=6,column=1)

```

Figure 185: Register window (xi)

```

227     radiobtn1=ttk.Radiobutton(class_student_frame,variable=self.var_radio1,text="Take Photo Sample",value="Yes")
228     radiobtn1.grid(row=6,column=1)
229
230     radiobtn2=ttk.Radiobutton(class_student_frame,variable=self.var_radio1,text="No Photo Sample",value="NO")
231     radiobtn2.grid(row=6,column=2)
232
233     #button frame
234     btn_frame=Frame(class_student_frame,bd=2,relief=RIDGE,bg="white")
235     btn_frame.place(x=0,y=235,width=655,height=80)
236
237     #Save Button
238     save_btn=Button(btn_frame,command=self.add_data,text="Save",width=18,font=("times new roman",12,"bold"),bg="Green",fg="white")
239     save_btn.grid(row=0,column=0)
240
241     #Update Button
242     update_btn=Button(btn_frame,command=self.update_data,text="Update",width=18,font=("times new roman",12,"bold"),bg="Green",fg="white")
243     update_btn.grid(row=0,column=1)
244
245     #Delete Button
246     delete_btn=Button(btn_frame,text="Delete",command=self.delete_data,width=18,font=("times new roman",12,"bold"),bg="Green",fg="white")
247     delete_btn.grid(row=0,column=2)
248
249     #Reset Button
250     reset_btn=Button(btn_frame,command=self.reset_data,text="Reset",width=18,font=("times new roman",12,"bold"),bg="Green",fg="white")
251     reset_btn.grid(row=0,column=3)
252
253     #frame for take photo and update photo
254     btn_frame1=Frame(class_student_frame,bd=2,relief=RIDGE)
255     btn_frame1.place(x=0,y=270,width=655,height=80)
256
257     take_photo_btn=Button(btn_frame1,command=self.generate_dataset,text="Take Photo Sample",width=70,font=("times new roman",13,"bold"),bg="green",fg="white")
258     take_photo_btn.grid(row=0,column=0)

```

Figure 186: Register window (xii)

```

283     Right_frame=LabelFrame(main_frame,bd=2,relief=RIDGE,text="Student Detail",font=("times new roman",14,"bold"))
284     Right_frame.place(x=680,y=10,width=735,height=600)
285
286     #Table frame
287     table_frame=Frame(Right_frame,bd=2,bg="white",relief=RIDGE)
288     table_frame.place(x=5,y=5,width=720,height=565)
289
290     #scroll bar
291     scroll_x=ttk.Scrollbar(table_frame,orient=HORIZONTAL)
292     scroll_y=ttk.Scrollbar(table_frame,orient=VERTICAL)
293
294     self.student_table=ttk.Treeview(table_frame,column=("dep","course","year","sem","id","name","div","roll","gender","dob","email","phone","address","teacher"))
295     self.student_table = ttk.Treeview(table_frame,columns=("dep", 'course', 'year', 'sem', 'id', 'name', 'div', 'roll', 'gender', 'dob', 'email', 'phone', 'address', 'teacher'))
296     scroll_x.pack(side=BOTTOM,fill=X)
297     scroll_y.pack(side=RIGHT,fill=Y)
298     scroll_x.config(command=self.student_table.xview)
299     scroll_y.config(command=self.student_table.yview)
300
301     self.student_table.heading("dep",text="Department")
302     self.student_table.heading("course",text="Course")
303     self.student_table.heading("year",text="Year")
304     self.student_table.heading("sem",text="Semester")
305     self.student_table.heading("id",text="Student ID")
306     self.student_table.heading("name",text="Name")
307     self.student_table.heading("div",text="Division")
308     self.student_table.heading("roll",text="Roll")
309     self.student_table.heading("gender",text="Gender")
310     self.student_table.heading("dob",text="DOB")
311     self.student_table.heading("email",text="Email")
312     self.student_table.heading("phone",text="Phone")
313     self.student_table.heading("address",text="Address")
314     self.student_table.heading("teacher",text="Teacher")
315     self.student_table.heading("photo",text="PhotoSampleStatus")

```

Figure 187: Register window (xiii)

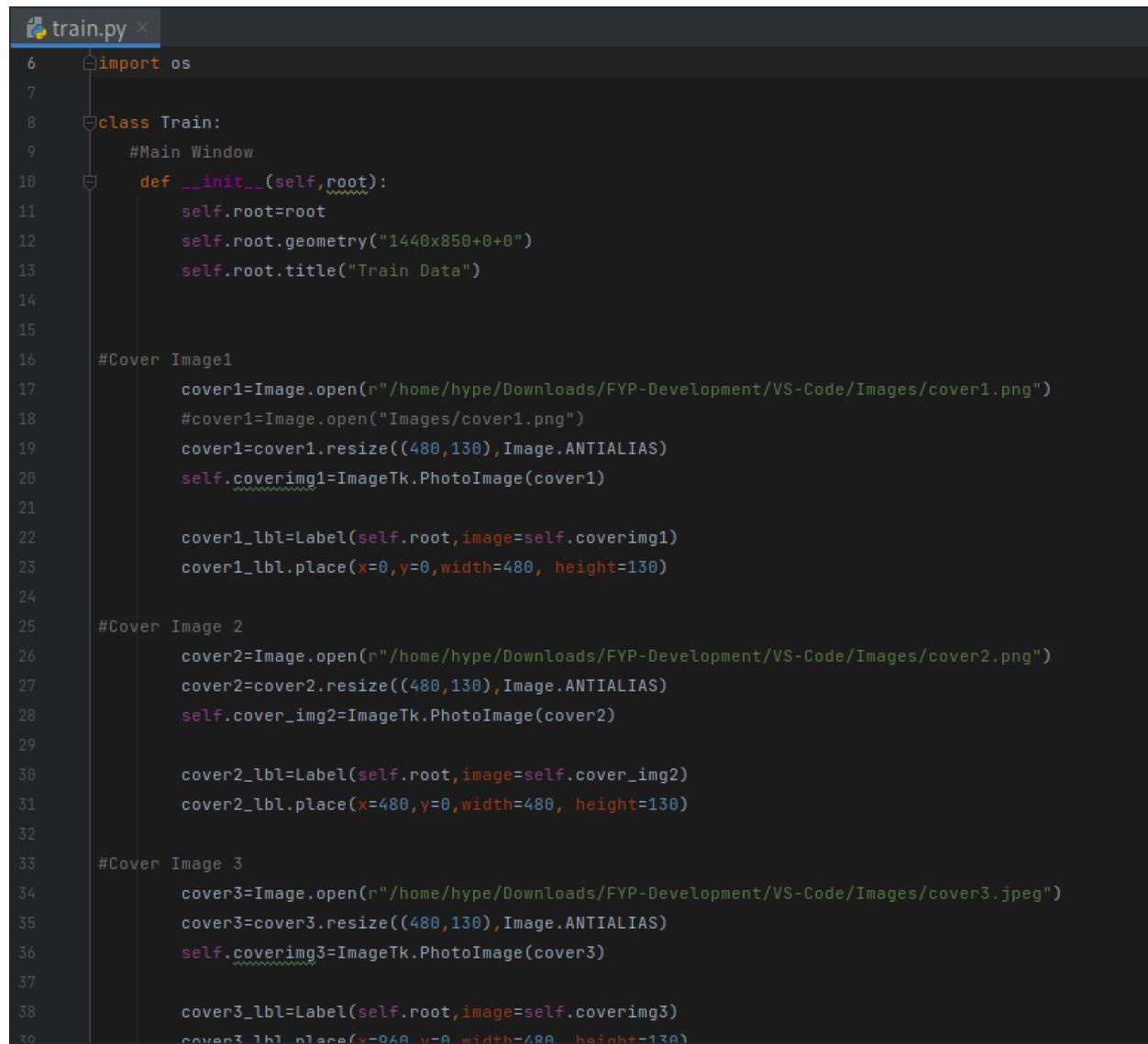
```

315     self.student_table.heading( "photo" ,text= "PhotoSamplesStatus" )
316     self.student_table[ "show" ]="headings"
317
318     self.student_table.column("dep",width=180)
319     self.student_table.column("course",width=90)
320     self.student_table.column("year",width=70)
321     self.student_table.column("sem",width=100)
322     self.student_table.column("id",width=100)
323     self.student_table.column("name",width=120)
324     self.student_table.column("roll",width=50)
325     self.student_table.column("gender",width=80)
326     self.student_table.column("div",width=50)
327     self.student_table.column("dob",width=100)
328     self.student_table.column("email",width=180)
329     self.student_table.column("phone",width=100)
330     self.student_table.column("address",width=100)
331     self.student_table.column("teacher",width=100)
332     self.student_table.column("photo",width=200)
333
334     self.student_table.pack(fill=BOTH,expand=1)
335     self.student_table.bind( "<ButtonRelease>" ,self.get_cursor)
336     self.fetch_data()

```

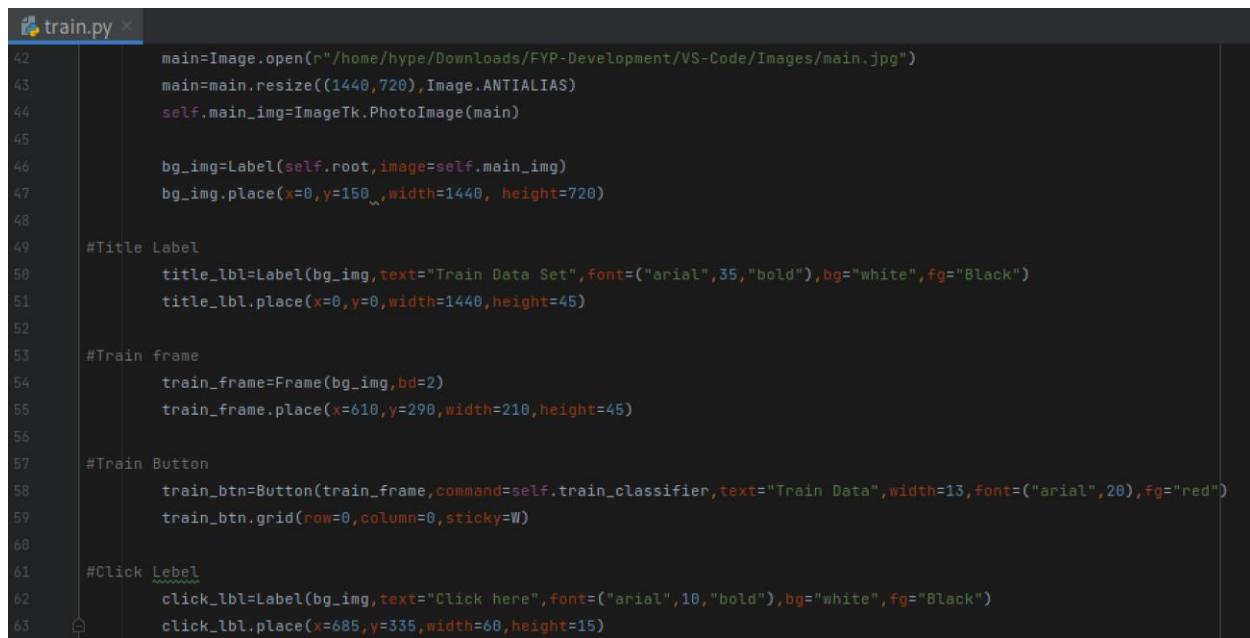
Figure 188: Register window (xiv)

#### 8.6.1.4 Train Window



```
train.py
6 import os
7
8 class Train:
9     #Main Window
10    def __init__(self,root):
11        self.root=root
12        self.root.geometry("1440x850+0+0")
13        self.root.title("Train Data")
14
15
16    #Cover Image1
17    cover1=Image.open(r"/home/hype/Downloads/FYP-Development/VS-Code/Images/cover1.png")
18    #cover1=Image.open("Images/cover1.png")
19    cover1=cover1.resize((480,130),Image.ANTIALIAS)
20    self.coverimg1=ImageTk.PhotoImage(cover1)
21
22    cover1_lbl=Label(self.root,image=self.coverimg1)
23    cover1_lbl.place(x=0,y=0,width=480, height=130)
24
25    #Cover Image 2
26    cover2=Image.open(r"/home/hype/Downloads/FYP-Development/VS-Code/Images/cover2.png")
27    cover2=cover2.resize((480,130),Image.ANTIALIAS)
28    self.cover_img2=ImageTk.PhotoImage(cover2)
29
30    cover2_lbl=Label(self.root,image=self.cover_img2)
31    cover2_lbl.place(x=480,y=0,width=480, height=130)
32
33    #Cover Image 3
34    cover3=Image.open(r"/home/hype/Downloads/FYP-Development/VS-Code/Images/cover3.jpeg")
35    cover3=cover3.resize((480,130),Image.ANTIALIAS)
36    self.coverimg3=ImageTk.PhotoImage(cover3)
37
38    cover3_lbl=Label(self.root,image=self.coverimg3)
39    cover3_lbl.place(x=960,y=0,width=480, height=130)
```

Figure 189: Train window (xv)



```
42     main=Image.open(r"/home/hype/Downloads/FYP-Development/VS-Code/Images/main.jpg")
43     main=main.resize((1440, 720),Image.ANTIALIAS)
44     self.main_img=ImageTk.PhotoImage(main)
45
46     bg_img=Label(self.root,image=self.main_img)
47     bg_img.place(x=0,y=150,width=1440, height=720)
48
49     #Title Label
50     title_lbl=Label(bg_img,text="Train Data Set",font=("arial",35,"bold"),bg="white",fg="Black")
51     title_lbl.place(x=0,y=0,width=1440,height=45)
52
53     #Train frame
54     train_frame=Frame(bg_img,bd=2)
55     train_frame.place(x=610,y=290,width=210,height=45)
56
57     #Train Button
58     train_btn=Button(train_frame,command=self.train_classifier,text="Train Data",width=13,font=("arial",20),fg="red")
59     train_btn.grid(row=0,column=0,sticky=W)
60
61     #Click Label
62     click_lbl=Label(bg_img,text="Click here",font=("arial",10,"bold"),bg="white",fg="Black")
63     click_lbl.place(x=685,y=335,width=60,height=15)
```

Figure 190: Train window (xvi)

### 8.6.1.5 Recognition Window

```

1  from tkinter import*
2  from PIL import Image,ImageTk
3  import mysql.connector
4  import cv2
5  from datetime import datetime
6
7  class face_recognition:
8      #Main Window
9      def __init__(self,root):
10         self.root=root
11         self.root.geometry("1440x850+0+0")
12         self.root.title("Face Recognition Attendance System")
13
14     #Cover Image1
15         cover1=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover1.png")
16         #cover1=Image.open("Images/cover1.png")
17         cover1=cover1.resize((480,130),Image.ANTIALIAS)
18         self.coverimg1=ImageTk.PhotoImage(cover1)
19
20         cover1_lbl=Label(self.root,image=self.coverimg1)
21         cover1_lbl.place(x=0,y=0,width=480, height=130)
22
23     #Cover Image 2
24         cover2=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover2.png")
25         cover2=cover2.resize((480,130),Image.ANTIALIAS)
26         self.cover_img2=ImageTk.PhotoImage(cover2)
27
28         cover2_lbl=Label(self.root,image=self.cover_img2)
29         cover2_lbl.place(x=480,y=0,width=480, height=130)
30
31     #Cover Image 3
32         cover3=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover3.jpeg")
33         cover3=cover3.resize((480,130),Image.ANTIALIAS)

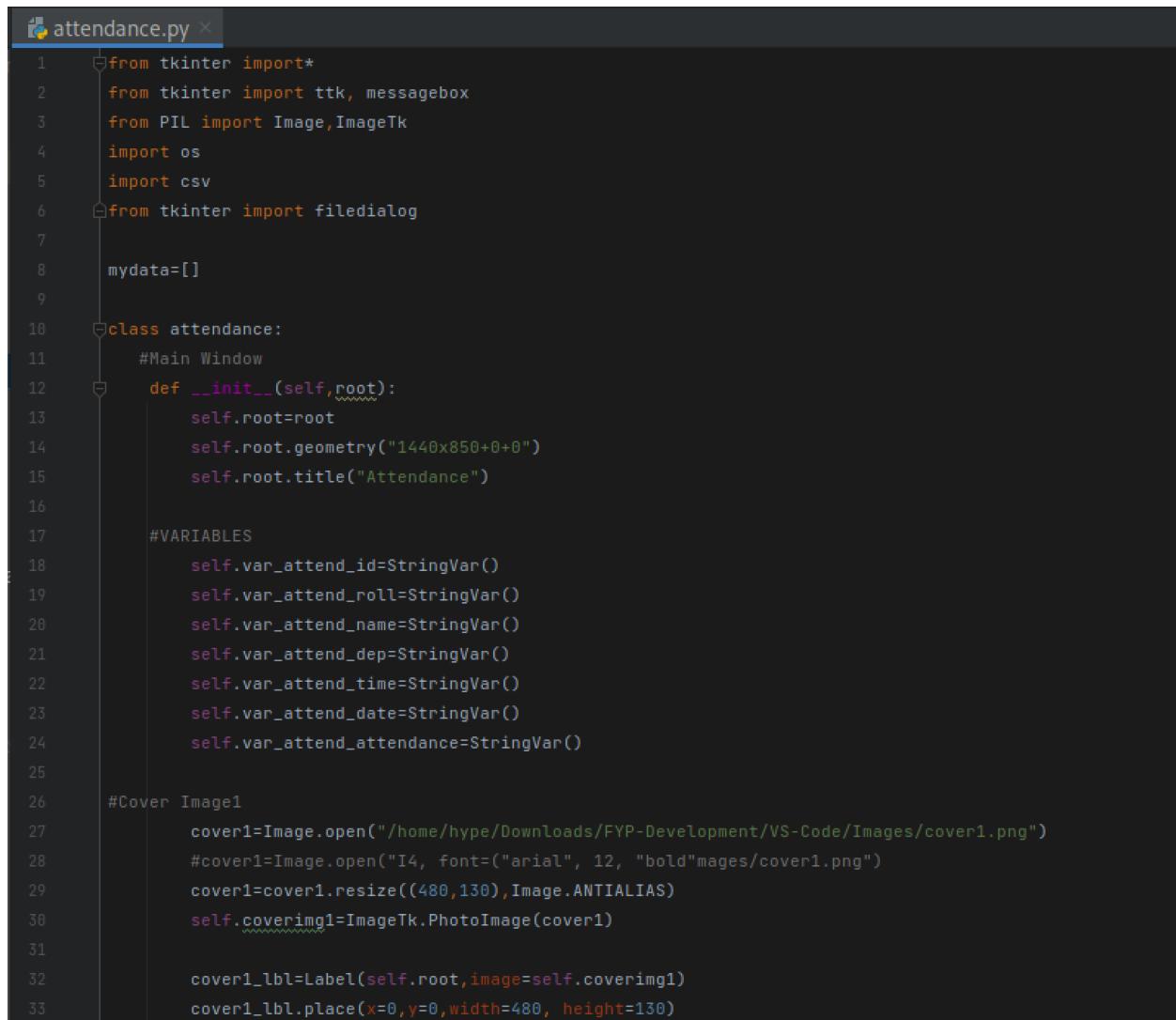
```

Figure 191: Recognition window (i)

```
 31 #Cover Image 3
 32     cover3=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover3.jpeg")
 33     cover3=cover3.resize((480,130),Image.ANTIALIAS)
 34     self.coverimg3=ImageTk.PhotoImage(cover3)
 35
 36     cover3_lbl=Label(self.root,image=self.coverimg3)
 37     cover3_lbl.place(x=960,y=0,width=480, height=130)
 38
 39 #Background Image
 40     main=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/FACE_RECOGNITION.jpg")
 41     main=main.resize((1440,720),Image.ANTIALIAS)
 42     self.main_img=ImageTk.PhotoImage(main)
 43
 44     bg_img=Label(self.root,image=self.main_img)
 45     bg_img.place(x=0,y=150,width=1440, height=720)
 46
 47 #Title Label
 48     title_lbl=Label(bg_img,text="Face Recognition",font=("arial",25,"bold"),bg="white",fg="Black")
 49     title_lbl.place(x=0,y=0,width=1440,height=45)
 50
 51 #Face Recognition frame
 52     train_frame=Frame(bg_img,bd=2)
 53     train_frame.place(x=610,y=490,width=210,height=45)
 54
 55 #Face Recognition Button
 56     train_btn=Button(train_frame,command=self.face_recog, text="Face Recognize", width=13, font=("arial",19,"bold"), fg='white')
 57     train_btn.grid(row=0,column=0,sticky=W)
 58
 59 #Click Label
 60     click_lbl=Label(bg_img,text="Click here",font=("arial",9,"bold"),bg="white",fg="red")
 61     click_lbl.place(x=685,y=535,width=60,height=15)
```

Figure 192: Recognition window (ii)

### 8.6.1.6 Attendance Window

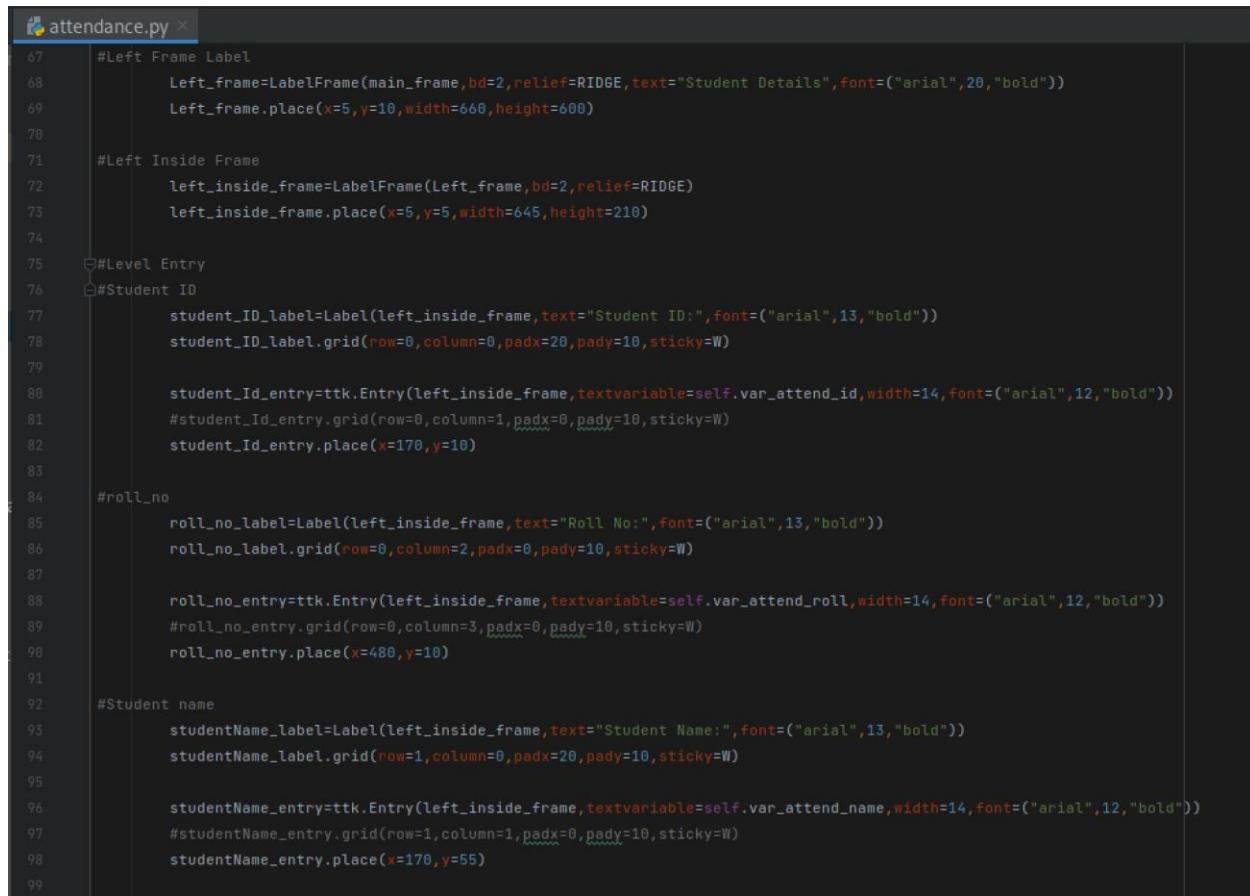


```
attendance.py
1  from tkinter import*
2  from tkinter import ttk, messagebox
3  from PIL import Image,ImageTk
4  import os
5  import csv
6  from tkinter import filedialog
7
8  mydata=[]
9
10 class attendance:
11     #Main Window
12     def __init__(self,root):
13         self.root=root
14         self.root.geometry("1440x850+0+0")
15         self.root.title("Attendance")
16
17     #VARIABLES
18     self.var_attend_id=StringVar()
19     self.var_attend_roll=StringVar()
20     self.var_attend_name=StringVar()
21     self.var_attend_dep=StringVar()
22     self.var_attend_time=StringVar()
23     self.var_attend_date=StringVar()
24     self.var_attend_attendance=StringVar()
25
26     #Cover Image1
27     cover1=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover1.png")
28     #cover1=Image.open("I4, font=("arial", 12, "bold")mages/cover1.png")
29     cover1=cover1.resize((480,130),Image.ANTIALIAS)
30     self.coverimg1=ImageTk.PhotoImage(cover1)
31
32     cover1_lbl=Label(self.root,image=self.coverimg1)
33     cover1_lbl.place(x=0,y=0,width=480, height=130)
```

Figure 193: Attendance window (i)

```
attendance.py
34
35     #Cover Image 2
36     cover2=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover2.png")
37     cover2=cover2.resize((480,130),Image.ANTIALIAS)
38     self.cover_img2=ImageTk.PhotoImage(cover2)
39
40     cover2_lbl=Label(self.root,image=self.cover_img2)
41     cover2_lbl.place(x=480,y=0,width=480, height=130)
42
43     #Cover Image 3
44     cover3=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/cover3.jpeg")
45     cover3=cover3.resize((480,130),Image.ANTIALIAS)
46     self.coverimg3=ImageTk.PhotoImage(cover3)
47
48     cover3_lbl=Label(self.root,image=self.coverimg3)
49     cover3_lbl.place(x=960,y=0,width=480, height=130)
50
51     #bg image
52     main=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/main.png")
53     main=main.resize((1440,720),Image.ANTIALIAS)
54     self.main_img=ImageTk.PhotoImage(main)
55
56     bg_img=Label(self.root,image=self.main_img)
57     bg_img.place(x=0,y=150, width=1440, height=720)
58
59     #Title Label
60     title_lbl=Label(bg_img,text="Student Attendance Detail",font=("arial",35,"bold"),bg="white",fg="Black")
61     title_lbl.place(x=0,y=0,width=1440,height=45)
62
63     #Main Frame
64     main_frame=Frame(bg_img,bd=2,bg="#988C89")
65     main_frame.place(x=0,y=50,width=1480,height=720)
```

Figure 194: Attendance window (ii)

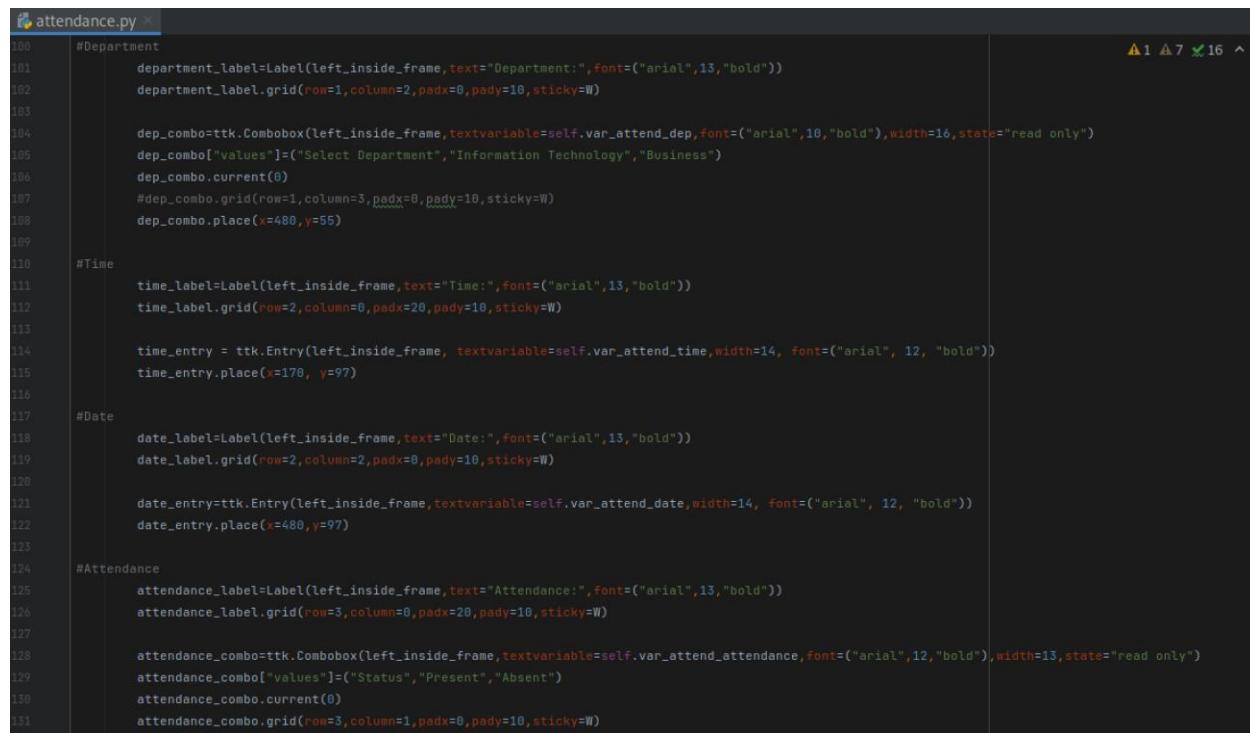


```

67     #Left Frame Label
68         Left_frame=LabelFrame(main_frame,bd=2,relief=RIDGE,text="Student Details",font=("arial",20,"bold"))
69         Left_frame.place(x=5,y=10,width=660,height=600)
70
71     #Left Inside Frame
72         left_inside_frame=LabelFrame(Left_frame,bd=2,relief=RIDGE)
73         left_inside_frame.place(x=5,y=5,width=645,height=210)
74
75     #Level Entry
76     #Student ID
77         student_ID_label=Label(left_inside_frame,text="Student ID:",font=("arial",13,"bold"))
78         student_ID_label.grid(row=0,column=0,padx=20,pady=10,sticky=W)
79
80         student_Id_entry=ttk.Entry(left_inside_frame,textvariable=self.var_attend_id,width=14,font=("arial",12,"bold"))
81         #student_Id_entry.grid(row=0,column=1,padx=0,pady=10,sticky=W)
82         student_Id_entry.place(x=170,y=10)
83
84     #roll_no
85         roll_no_label=Label(left_inside_frame,text="Roll No:",font=("arial",13,"bold"))
86         roll_no_label.grid(row=0,column=2,padx=0,pady=10,sticky=W)
87
88         roll_no_entry=ttk.Entry(left_inside_frame,textvariable=self.var_attend_roll,width=14,font=("arial",12,"bold"))
89         #roll_no_entry.grid(row=0,column=3,padx=0,pady=10,sticky=W)
90         roll_no_entry.place(x=480,y=10)
91
92     #Student name
93         studentName_label=Label(left_inside_frame,text="Student Name:",font=("arial",13,"bold"))
94         studentName_label.grid(row=1,column=0,padx=20,pady=10,sticky=W)
95
96         studentName_entry=ttk.Entry(left_inside_frame,textvariable=self.var_attend_name,width=14,font=("arial",12,"bold"))
97         #studentName_entry.grid(row=1,column=1,padx=0,pady=10,sticky=W)
98         studentName_entry.place(x=170,y=55)
99

```

Figure 195: Attendance window (iii)

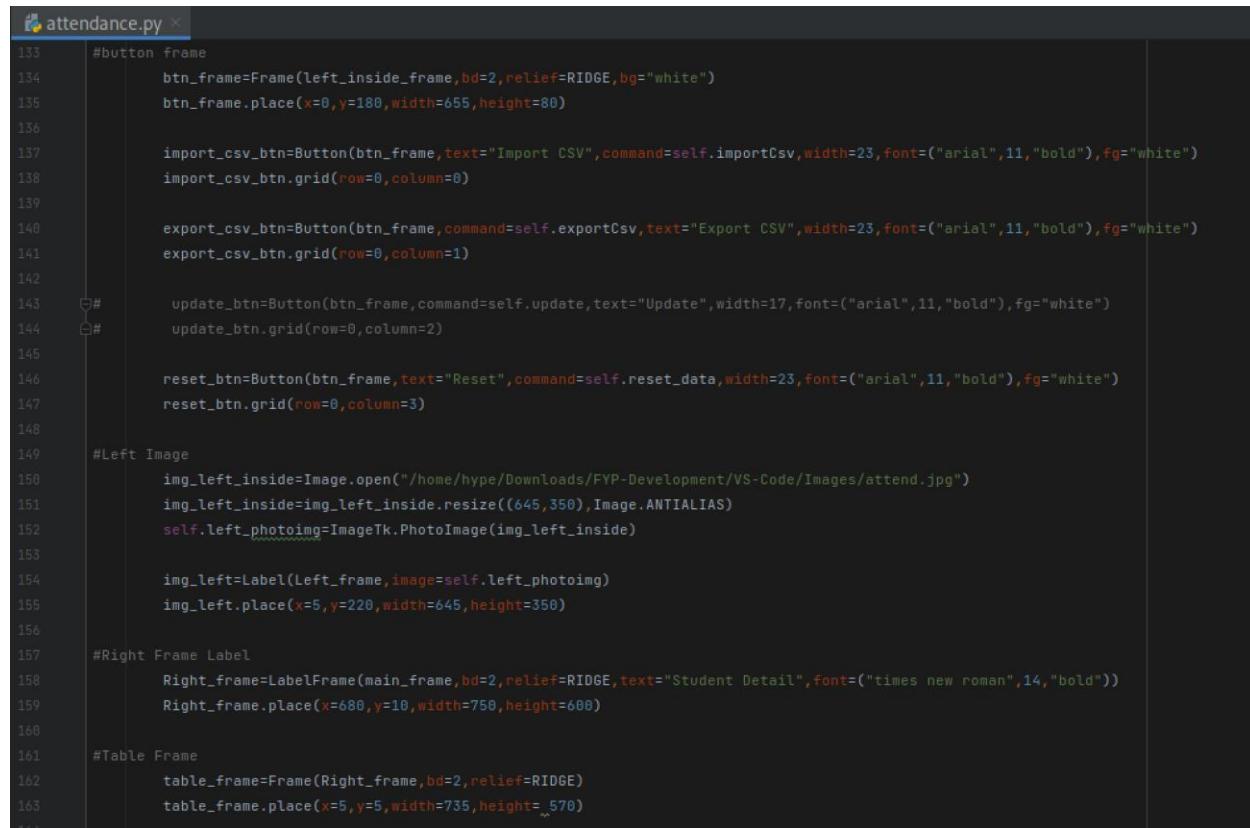


```

100     #Department
101         department_label=Label(left_inside_frame,text="Department:",font=("arial",13,"bold"))
102         department_label.grid(row=1,column=2,padx=0,pady=10,sticky=W)
103
104         dep_combo=ttk.Combobox(left_inside_frame,textvariable=self.var_attend_dep,font=("arial",10,"bold"),width=16,state="read only")
105         dep_combo["values"]=(("Select Department", "Information Technology", "Business"))
106         dep_combo.current(0)
107         #dep_combo.grid(row=1,column=3,padx=0,pady=10,sticky=W)
108         dep_combo.place(x=480,y=55)
109
110     #Time
111         time_label=Label(left_inside_frame,text="Time:",font=("arial",13,"bold"))
112         time_label.grid(row=2,column=0,padx=20,pady=10,sticky=W)
113
114         time_entry = ttk.Entry(left_inside_frame, textvariable=self.var_attend_time, width=14, font=("arial", 12, "bold"))
115         time_entry.place(x=170, y=97)
116
117     #Date
118         date_label=Label(left_inside_frame,text="Date:",font=("arial",13,"bold"))
119         date_label.grid(row=2,column=2,padx=0,pady=10,sticky=W)
120
121         date_entry=ttk.Entry(left_inside_frame, textvariable=self.var_attend_date, width=14, font=("arial", 12, "bold"))
122         date_entry.place(x=480,y=97)
123
124     #Attendance
125         attendance_label=Label(left_inside_frame,text="Attendance:",font=("arial",13,"bold"))
126         attendance_label.grid(row=3,column=0,padx=20,pady=10,sticky=W)
127
128         attendance_combo=ttk.Combobox(left_inside_frame, textvariable=self.var_attend_attendance,font=("arial",12,"bold"),width=13,state="read only")
129         attendance_combo["values"]=(("Status", "Present", "Absent"))
130         attendance_combo.current(0)
131         attendance_combo.grid(row=3,column=1,padx=0,pady=10,sticky=W)

```

Figure 196: Attendance window (iv)

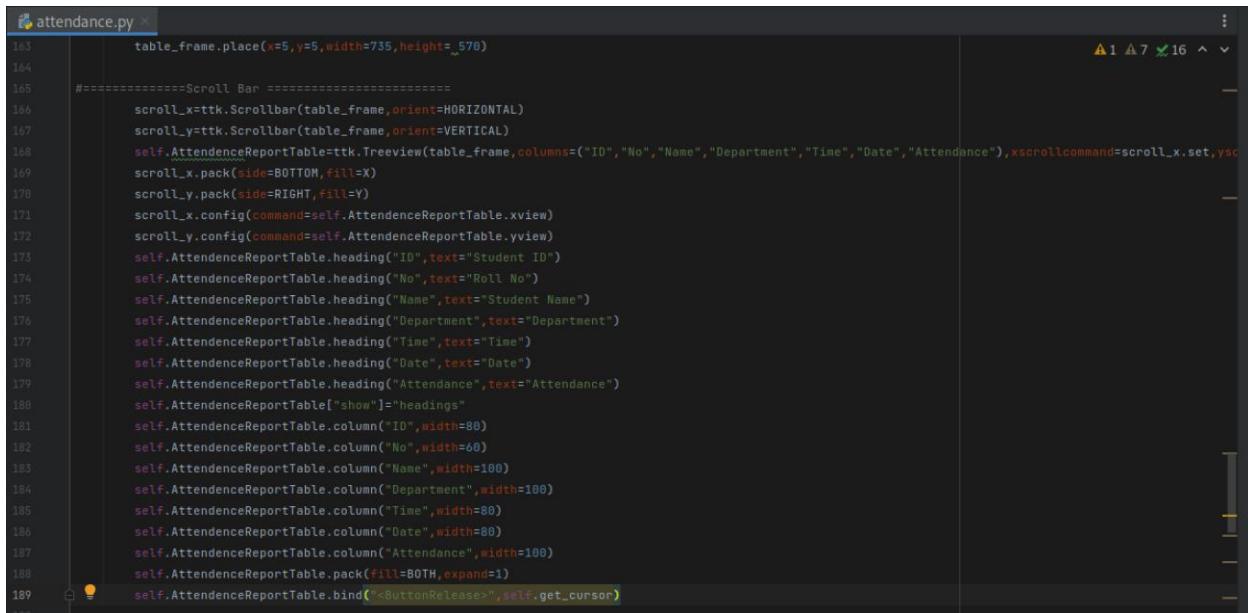


```

133     #button frame
134         btn_frame=Frame(left_inside_frame,bd=2,relief=RIDGE,bg="white")
135         btn_frame.place(x=0,y=180,width=655,height=80)
136
137         import_csv_btn=Button(btn_frame,text="Import CSV",command=self.importCsv,width=23,font=("arial",11,"bold"),fg="white")
138         import_csv_btn.grid(row=0,column=0)
139
140         export_csv_btn=Button(btn_frame,command=self.exportCsv,text="Export CSV",width=23,font=("arial",11,"bold"),fg="white")
141         export_csv_btn.grid(row=0,column=1)
142
143         update_btn=Button(btn_frame,command=self.update,text="Update",width=17,font=("arial",11,"bold"),fg="white")
144         update_btn.grid(row=0,column=2)
145
146         reset_btn=Button(btn_frame,text="Reset",command=self.reset_data,width=23,font=("arial",11,"bold"),fg="white")
147         reset_btn.grid(row=0,column=3)
148
149     #Left Image
150         img_left_inside=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/attend.jpg")
151         img_left_inside=img_left_inside.resize((645,350),Image.ANTIALIAS)
152         self.left_photoimg=ImageTk.PhotoImage(img_left_inside)
153
154         img_left=Label(left_frame,image=self.left_photoimg)
155         img_left.place(x=5,y=220,width=645,height=350)
156
157     #Right Frame Label
158         Right_frame=LabelFrame(main_frame,bd=2,relief=RIDGE,text="Student Detail",font=("times new roman",14,"bold"))
159         Right_frame.place(x=680,y=10,width=750,height=600)
160
161     #Table Frame
162         table_frame=Frame(Right_frame,bd=2,relief=RIDGE)
163         table_frame.place(x=5,y=5,width=735,height=570)

```

Figure 197: Attendance window (v)

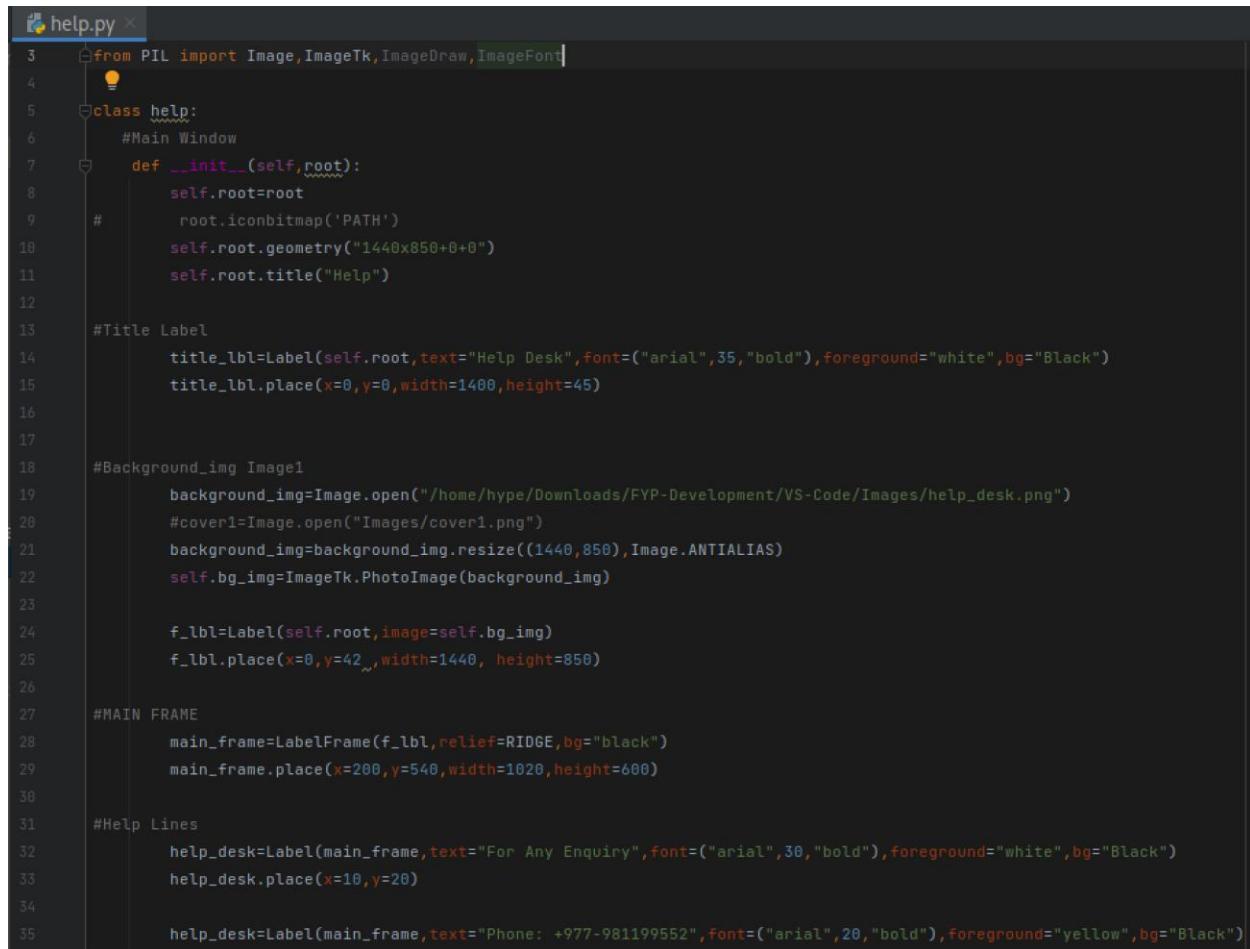


The screenshot shows a terminal window with the file name 'attendance.py' at the top. The code is a Python script using the Tkinter library to create a window titled 'Attendance'. The window contains a treeview widget ('AttendanceReportTable') with columns for ID, No, Name, Department, Time, Date, and Attendance. The treeview is placed in a frame at coordinates (5, 5) with width 735 and height 570. Horizontal and vertical scrollbars are attached to the treeview. The code includes logic for setting column widths, configuring headings, and binding a cursor event to the treeview.

```
163     table_frame.place(x=5,y=5,width=735,height=570)
164
165 #=====Scroll Bar =====
166 scroll_x=ttk.Scrollbar(table_frame,orient=HORIZONTAL)
167 scroll_y=ttk.Scrollbar(table_frame,orient=VERTICAL)
168 self.AttendanceReportTable=ttk.Treeview(table_frame,columns=("ID","No","Name","Department","Time","Date","Attendance"),xscrollcommand=scroll_x.set,ysc
169 scroll_x.pack(side=BOTTOM,fill=X)
170 scroll_y.pack(side=RIGHT,fill=Y)
171 scroll_x.config(command=self.AttendanceReportTable.xview)
172 scroll_y.config(command=self.AttendanceReportTable.yview)
173 self.AttendanceReportTable.heading("ID",text="Student ID")
174 self.AttendanceReportTable.heading("No",text="Roll No")
175 self.AttendanceReportTable.heading("Name",text="Student Name")
176 self.AttendanceReportTable.heading("Department",text="Department")
177 self.AttendanceReportTable.heading("Time",text="Time")
178 self.AttendanceReportTable.heading("Date",text="Date")
179 self.AttendanceReportTable.heading("Attendance",text="Attendance")
180 self.AttendanceReportTable["show"]="headings"
181 self.AttendanceReportTable.column("ID",width=80)
182 self.AttendanceReportTable.column("No",width=60)
183 self.AttendanceReportTable.column("Name",width=100)
184 self.AttendanceReportTable.column("Department",width=100)
185 self.AttendanceReportTable.column("Time",width=80)
186 self.AttendanceReportTable.column("Date",width=80)
187 self.AttendanceReportTable.column("Attendance",width=100)
188 self.AttendanceReportTable.pack(fill=BOTH,expand=1)
189 self.AttendanceReportTable.bind("<ButtonRelease>",self.get_cursor)
```

Figure 198: Attendance window (vi)

### 8.6.1.7 Help Desk window

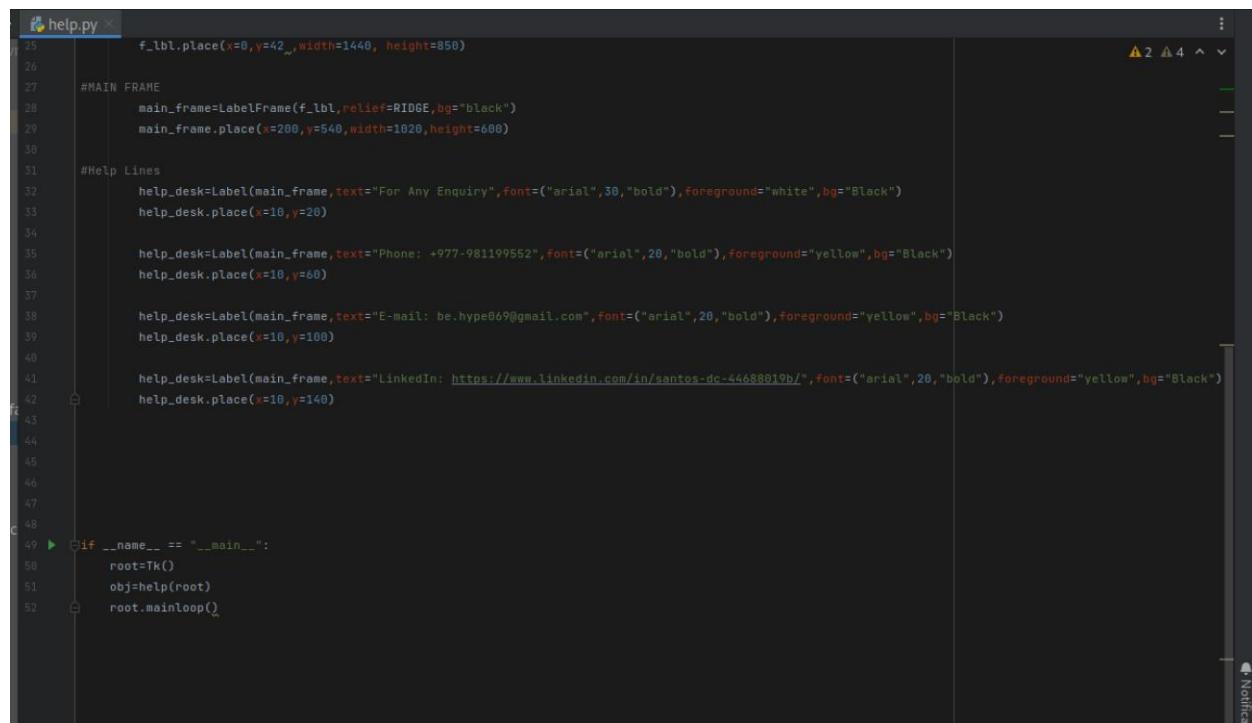


```

1  help.py
2
3  from PIL import Image,ImageTk,ImageDraw,ImageFont
4
5  class help:
6      #Main Window
7      def __init__(self,root):
8          self.root=root
9          #root.iconbitmap('PATH')
10         self.root.geometry("1440x850+0+0")
11         self.root.title("Help")
12
13     #Title Label
14     title_lbl=Label(self.root,text="Help Desk",font=("arial",35,"bold"),foreground="white",bg="Black")
15     title_lbl.place(x=0,y=0,width=1400,height=45)
16
17
18     #Background_img Image1
19     background_img=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/help_desk.png")
20     #cover1=Image.open("Images/cover1.png")
21     background_img=background_img.resize((1440,850),Image.ANTIALIAS)
22     self.bg_img=ImageTk.PhotoImage(background_img)
23
24     f_lbl=Label(self.root,image=self.bg_img)
25     f_lbl.place(x=0,y=42,width=1440, height=850)
26
27     #MAIN FRAME
28     main_frame=LabelFrame(f_lbl,relief=RIDGE,bg="black")
29     main_frame.place(x=200,y=540,width=1020,height=600)
30
31     #Help Lines
32     help_desk=Label(main_frame,text="For Any Enquiry",font=("arial",30,"bold"),foreground="white",bg="Black")
33     help_desk.place(x=10,y=20)
34
35     help_desk=Label(main_frame,text="Phone: +977-981199552",font=("arial",20,"bold"),foreground="yellow",bg="Black")

```

Figure 199: Help Desk Window (i)

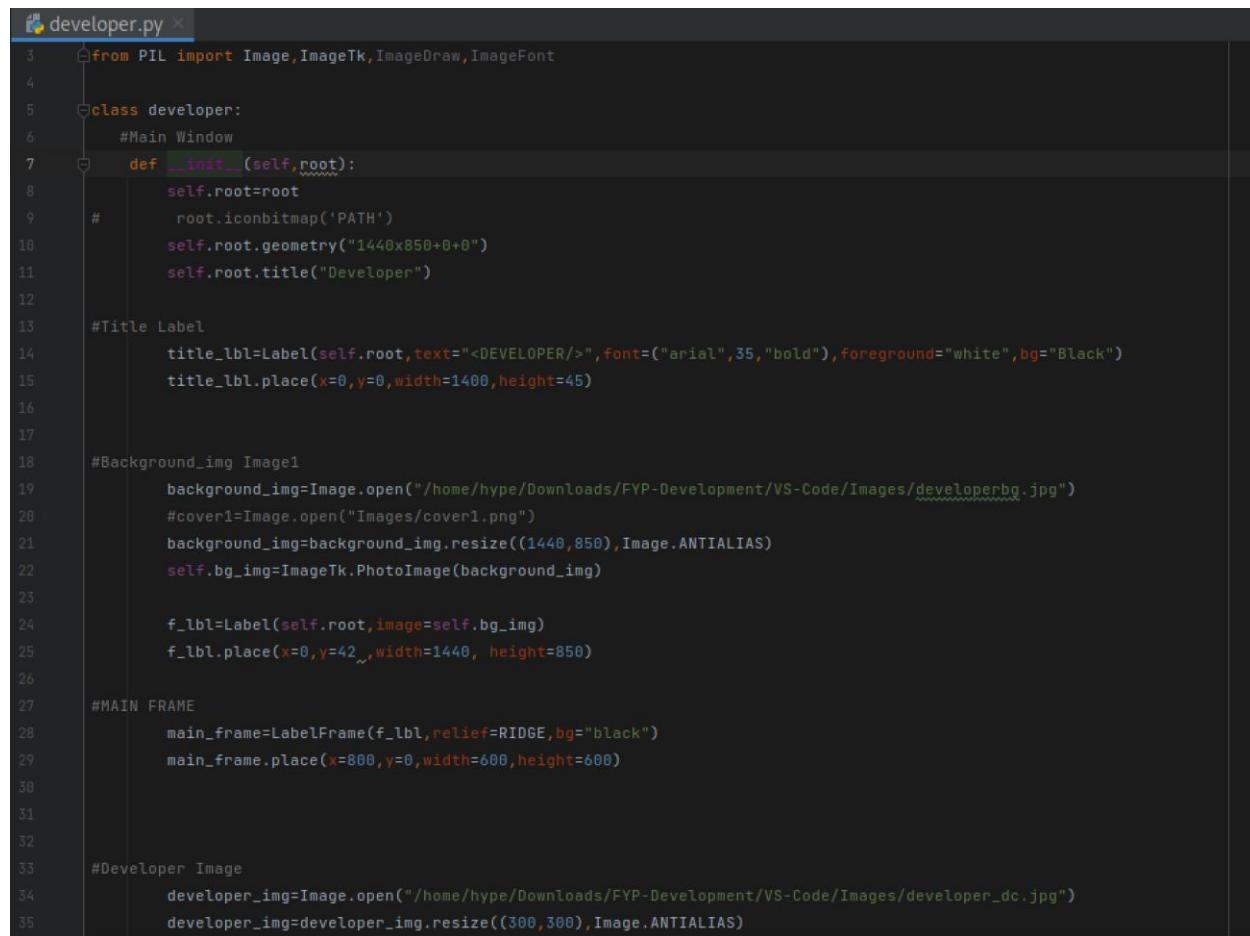


The screenshot shows a code editor window with a dark theme. The file is named 'help.py'. The code is a Python script for a Tkinter application. It starts by creating a main frame and placing a label at the top. Then, it defines several 'help\_desk' labels with specific text, font, foreground color (yellow), and background color (black). Finally, it checks if the script is run as the main module and creates a root window, an instance of the 'help' class, and enters the main loop.

```
 25     f_lbl.place(x=0,y=42,width=1440, height=850)
 26
 27 #MAIN FRAME
 28     main_frame=LabelFrame(f_lbl,relief=RIDGE,bg="black")
 29     main_frame.place(x=200,y=540,width=1020,height=600)
 30
 31 #Help Lines
 32     help_desk=Label(main_frame,text="For Any Enquiry",font=("arial",30,"bold"),foreground="white",bg="Black")
 33     help_desk.place(x=10,y=20)
 34
 35     help_desk=Label(main_frame,text="Phone: +977-981199552",font=("arial",20,"bold"),foreground="yellow",bg="Black")
 36     help_desk.place(x=10,y=60)
 37
 38     help_desk=Label(main_frame,text="E-mail: be.hype069@gmail.com",font=("arial",20,"bold"),foreground="yellow",bg="Black")
 39     help_desk.place(x=10,y=100)
 40
 41     help_desk=Label(main_frame,text="LinkedIn: https://www.linkedin.com/in/santos-dc-44688019b/",font=("arial",20,"bold"),foreground="yellow",bg="Black")
 42     help_desk.place(x=10,y=140)
 43
 44
 45
 46
 47
 48
 49 ▶ 50 if __name__ == "__main__":
 51     root=Tk()
 52     obj=help(root)
 53     root.mainloop()
```

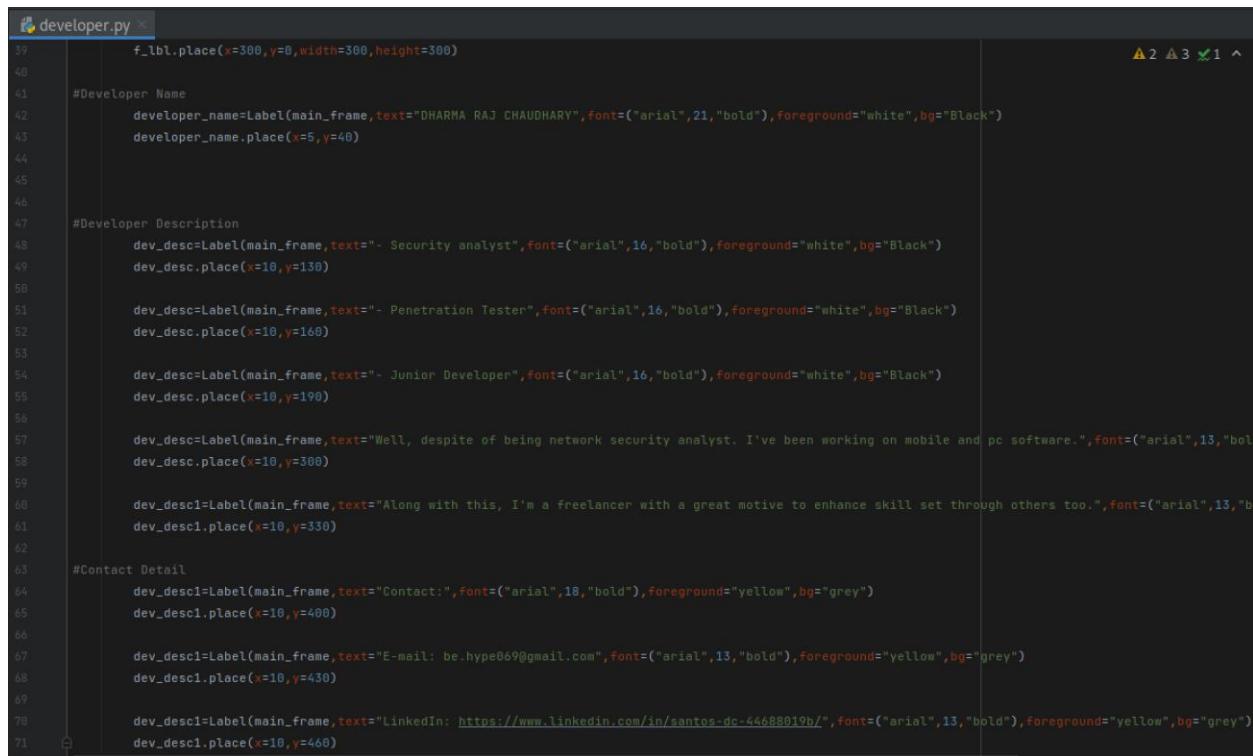
Figure 200: Help Desk Window (ii)

### 8.6.1.8 Developer Window



```
developer.py
from PIL import Image, ImageTk, ImageDraw, ImageFont
class developer:
    #Main Window
    def __init__(self,root):
        self.root=root
        #root.iconbitmap('PATH')
        self.root.geometry("1440x850+0+0")
        self.root.title("Developer")
    #Title Label
    title_lbl=Label(self.root, text="<DEVELOPER/>", font=("arial", 36, "bold"), foreground="white", bg="Black")
    title_lbl.place(x=0, y=0, width=1400, height=45)
    #Background_img Image1
    background_img=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/developerbg.jpg")
    #cover1=Image.open("Images/cover1.png")
    background_img=background_img.resize((1440, 850), Image.ANTIALIAS)
    self.bg_img=ImageTk.PhotoImage(background_img)
    f_lbl=Label(self.root, image=self.bg_img)
    f_lbl.place(x=0, y=42, width=1440, height=850)
    #MAIN FRAME
    main_frame=LabelFrame(f_lbl, relief=RIDGE, bg="black")
    main_frame.place(x=800, y=0, width=600, height=600)
    #Developer Image
    developer_img=Image.open("/home/hype/Downloads/FYP-Development/VS-Code/Images/developer_dc.jpg")
    developer_img=developer_img.resize((300, 300), Image.ANTIALIAS)
```

Figure 201: Developer window (i)



```

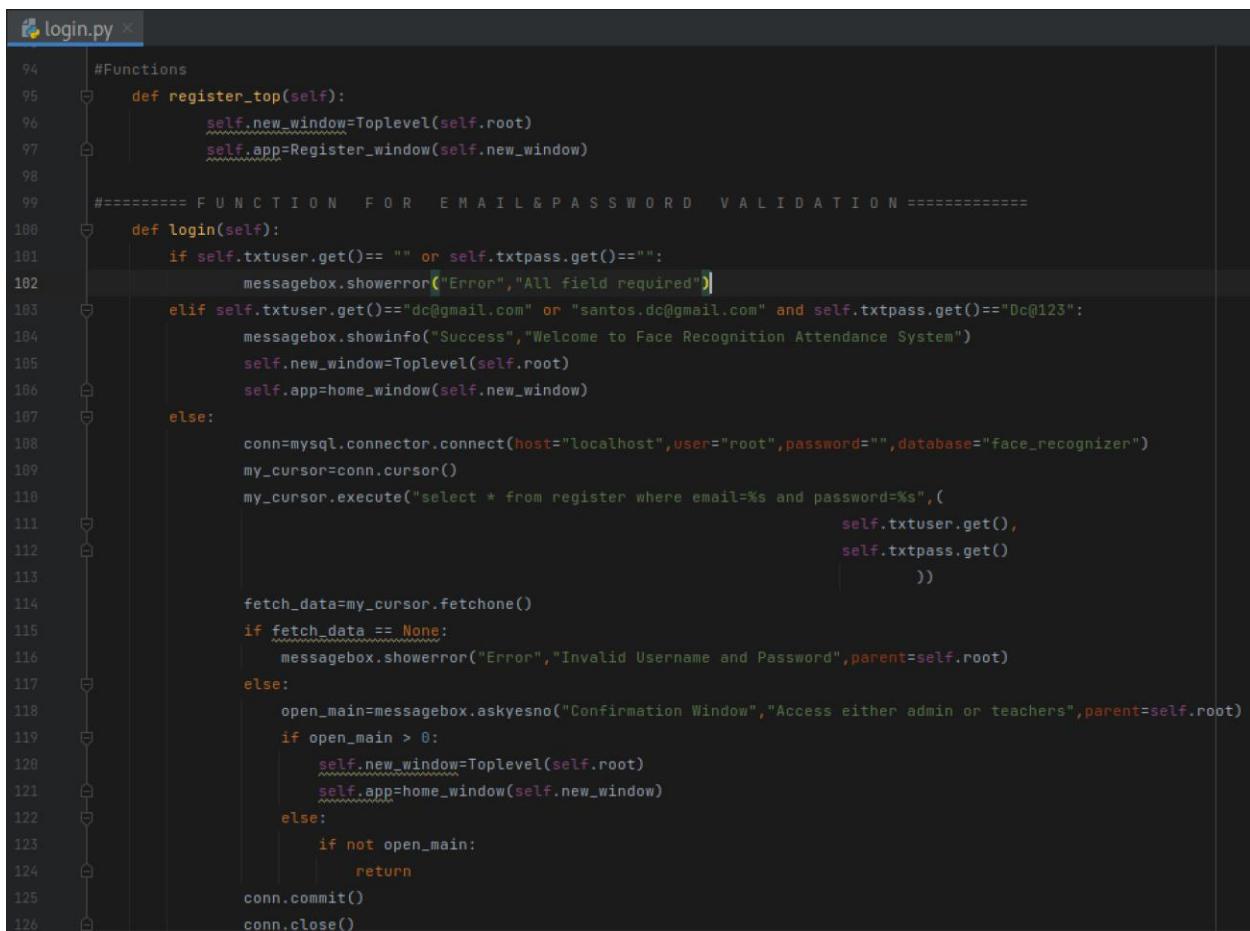
39     f_lbl.place(x=300,y=0,width=300,height=300)
40
41 #Developer Name
42     developer_name=Label(main_frame,text="DHARMA RAJ CHAUDHARY",font=("arial",21,"bold"),foreground="white",bg="Black")
43     developer_name.place(x=5,y=40)
44
45
46 #Developer Description
47     dev_desc=Label(main_frame,text="- Security analyst",font=("arial",16,"bold"),foreground="white",bg="Black")
48     dev_desc.place(x=10,y=130)
49
50     dev_desc=Label(main_frame,text="- Penetration Tester",font=("arial",16,"bold"),foreground="white",bg="Black")
51     dev_desc.place(x=10,y=160)
52
53     dev_desc=Label(main_frame,text="- Junior Developer",font=("arial",16,"bold"),foreground="white",bg="Black")
54     dev_desc.place(x=10,y=190)
55
56     dev_desc=Label(main_frame,text="Well, despite of being network security analyst. I've been working on mobile and pc software.",font=("arial",18,"bold"))
57     dev_desc.place(x=10,y=300)
58
59     dev_desc1=Label(main_frame,text="Along with this, I'm a freelancer with a great motive to enhance skill set through others too.",font=("arial",13,"bold"))
60     dev_desc1.place(x=10,y=330)
61
62 #Contact Detail
63     dev_desc1=Label(main_frame,text="Contact:",font=("arial",18,"bold"),foreground="yellow",bg="grey")
64     dev_desc1.place(x=10,y=400)
65
66     dev_desc1=Label(main_frame,text="E-mail: be.hype069@gmail.com",font=("arial",13,"bold"),foreground="yellow",bg="grey")
67     dev_desc1.place(x=10,y=430)
68
69     dev_desc1=Label(main_frame,text="LinkedIn: https://www.linkedin.com/in/santos-dc-44688019b/",font=("arial",13,"bold"),foreground="yellow",bg="grey")
70     dev_desc1.place(x=10,y=460)
71

```

Figure 202:Developer window (ii)

## 8.6.2 Sample Code for The Automation Script

### 8.6.2.1 Login



```

94     #Functions
95     def register_top(self):
96         self.new_window=Toplevel(self.root)
97         self.app=Register_window(self.new_window)
98
99     ##### F U N C T I O N   F O R   E M A I L & P A S S W O R D   V A L I D A T I O N #####
100    def login(self):
101        if self.txtuser.get() == "" or self.txtpass.get() == "":
102            messagebox.showerror("Error", "All field required")
103        elif self.txtuser.get() == "dc@gmail.com" or " santos.dc@gmail.com" and self.txtpass.get() == "Dc@123":
104            messagebox.showinfo("Success", "Welcome to Face Recognition Attendance System")
105            self.new_window=Toplevel(self.root)
106            self.app=home_window(self.new_window)
107        else:
108            conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
109            my_cursor=conn.cursor()
110            my_cursor.execute("select * from register where email=%s and password=%s",
111                                (self.txtuser.get(),
112                                 self.txtpass.get()
113                                 ))
114            fetch_data=my_cursor.fetchone()
115            if fetch_data == None:
116                messagebox.showerror("Error", "Invalid Username and Password",parent=self.root)
117            else:
118                open_main=messagebox.askyesno("Confirmation Window", "Access either admin or teachers",parent=self.root)
119                if open_main > 0:
120                    self.new_window=Toplevel(self.root)
121                    self.app=home_window(self.new_window)
122                else:
123                    if not open_main:
124                        return
125            conn.commit()
126            conn.close()

```

Figure 203: Login Window (i)

```

128     ===== R E S E T   P A S S W O R D   F U N C T I O N =====
129
130     def reset_pass(self):
131         if self.combo_security.get()=="Select":
132             messagebox.showerror("Error","Select the security question",parent=self.root2)
133         elif self.security_ans_entry.get()=="":
134             messagebox.showerror("Error","Please enter the answer",parent=self.root2)
135         elif self.new_password_entry.get()=="":
136             messagebox.showerror("Error","Please Enter new password",parent=self.root2)
137         else:
138             conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
139             my_cursor=conn.cursor()
140             query=("SELECT * FROM register WHERE email=%s AND SecurityQuestion=%s AND SecurityAnswer=%s")
141             value=(self.txtuser.get(),self.combo_security.get(),self.security_ans_entry.get())
142             my_cursor.execute(query,value)
143             fetch_data=my_cursor.fetchone()
144             if fetch_data==None:
145                 messagebox.showerror("Error","Please enter correct answer",parent=self.root2)
146             else:
147                 query=("UPDATE register SET password=%s WHERE email=%s")
148                 value=(self.new_password_entry.get(),self.txtuser.get())
149                 my_cursor.execute(query,value)
150                 messagebox.showinfo("Info","New password set, login now",parent=self.root2)
151                 conn.commit()
152                 conn.close()
153                 self.root2.destroy()

```

Figure 204: Login Window (ii)

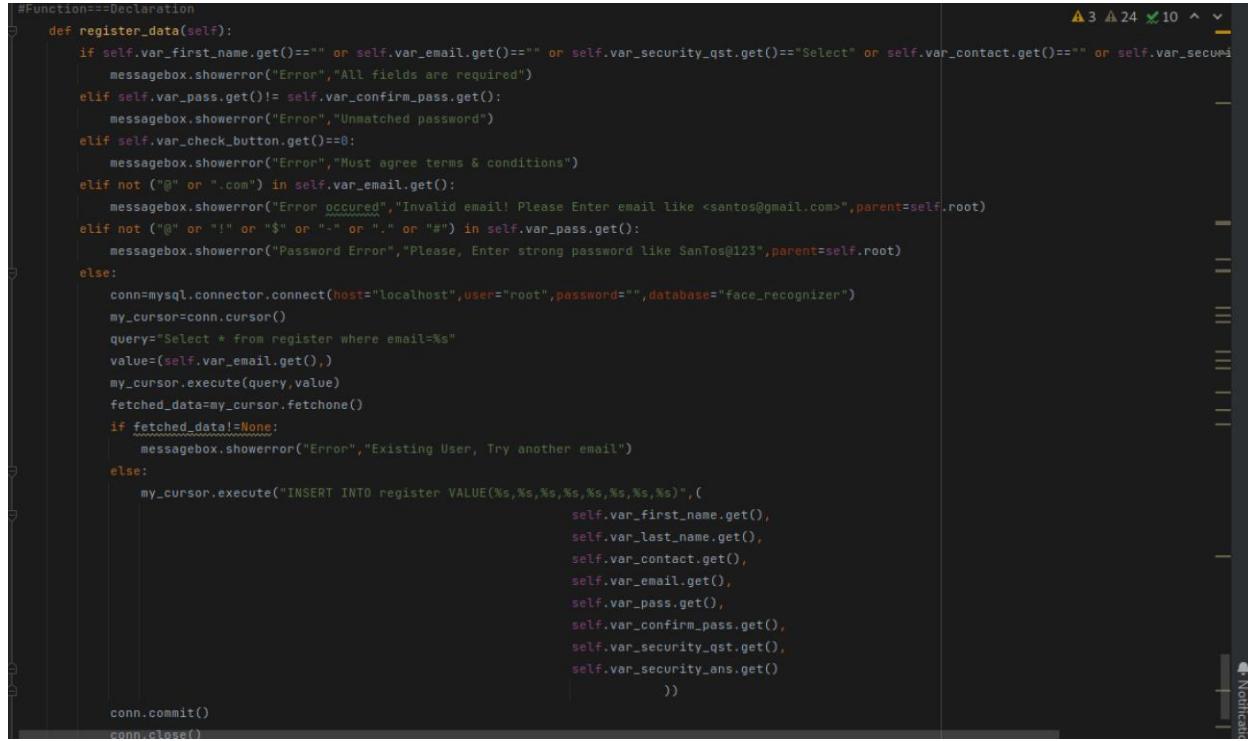
```

===== F O R G E T   P A S S W O R D   F U N C T I O N =====
1     def forgot_password_window(self):
2         if self.txtuser.get()=="":
3             messagebox.showerror("Alert","Enter email section to reset password")
4         else:
5             conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
6             my_cursor=conn.cursor()
7             query=("SELECT * FROM register WHERE email=%s")
8             value=(self.txtuser.get(),)
9             my_cursor.execute(query,value)
10            fetch_data=my_cursor.fetchone()
11            if fetch_data==None:
12                messagebox.showerror("Error","Email or Username not Found")
13            else:
14                conn.close()
15                self.root2=Toplevel()
16                self.root2.title("Forgot Password")
17                self.root2.geometry("400x420+400+120")

```

Figure 205: Login Window (iii)

### 8.6.2.2 Register Window



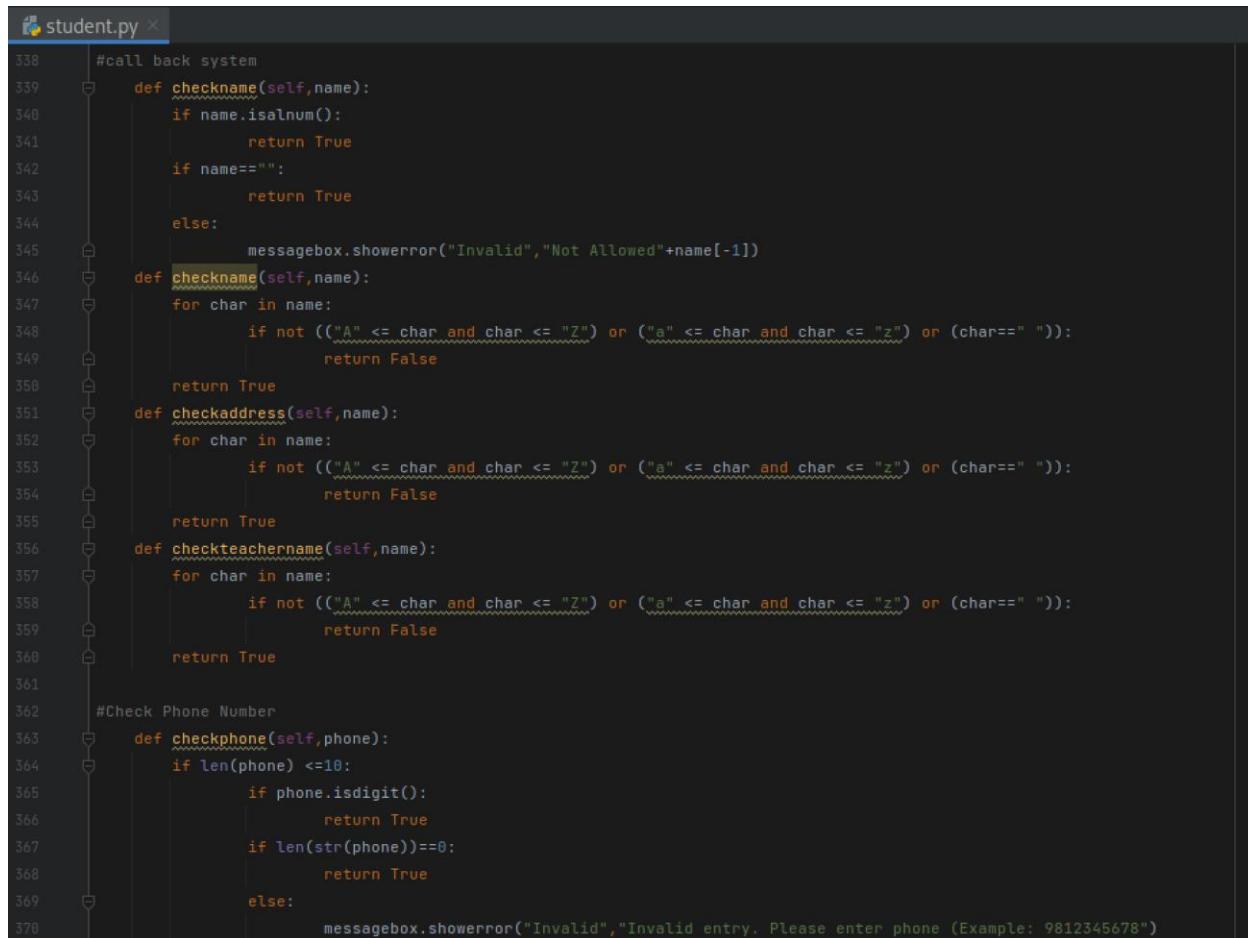
```

#Function==Declaration
def register_data(self):
    if self.var_first_name.get()=="" or self.var_email.get()=="" or self.var_security_qst.get()=="Select" or self.var_contact.get()=="" or self.var_secans.get()=="":
        messagebox.showerror("Error","All fields are required")
    elif self.var_pass.get()!= self.var_confirm_pass.get():
        messagebox.showerror("Error","Unmatched password")
    elif self.var_check_button.get()==0:
        messagebox.showerror("Error","Must agree terms & conditions")
    elif not ("@" or "!" or "$" or "-" or "." or "#") in self.var_email.get():
        messagebox.showerror("Error occurred","Invalid email! Please Enter email like <santos@gmail.com>",parent=self.root)
    elif not ("@" or "!" or "$" or "-" or "." or "#") in self.var_pass.get():
        messagebox.showerror("Password Error","Please, Enter strong password like SanTos@123",parent=self.root)
    else:
        conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
        my_cursor=conn.cursor()
        query="Select * from register where email=%s"
        value=(self.var_email.get(),)
        my_cursor.execute(query,value)
        fetched_data=my_cursor.fetchone()
        if fetched_data!=None:
            messagebox.showerror("Error","Existing User, Try another email")
        else:
            my_cursor.execute("INSERT INTO register VALUE(%s,%s,%s,%s,%s,%s,%s)",(
                self.var_first_name.get(),
                self.var_last_name.get(),
                self.var_contact.get(),
                self.var_email.get(),
                self.var_pass.get(),
                self.var_confirm_pass.get(),
                self.var_security_qst.get(),
                self.var_security_ans.get()
            ))
    conn.commit()
    conn.close()

```

Figure 206: Register window (i)

### 8.6.2.3 Student Window



```

338     #call back system
339     def checkname(self,name):
340         if name.isalnum():
341             return True
342         if name=="":
343             return True
344         else:
345             messagebox.showerror("Invalid","Not Allowed"+name[-1])
346     def checkname(self,name):
347         for char in name:
348             if not ((ord("A") <= ord(char) <= ord("Z")) or (ord("a") <= ord(char) <= ord("z")) or (char==" ")):
349                 return False
350         return True
351     def checkaddress(self,name):
352         for char in name:
353             if not ((ord("A") <= ord(char) <= ord("Z")) or (ord("a") <= ord(char) <= ord("z")) or (char==" ")):
354                 return False
355         return True
356     def checkteachername(self,name):
357         for char in name:
358             if not ((ord("A") <= ord(char) <= ord("Z")) or (ord("a") <= ord(char) <= ord("z")) or (char==" ")):
359                 return False
360         return True
361
362     #Check Phone Number
363     def checkphone(self,phone):
364         if len(phone) <=10:
365             if phone.isdigit():
366                 return True
367             if len(str(phone))==0:
368                 return True
369             else:
370                 messagebox.showerror("Invalid","Invalid entry. Please enter phone (Example: 9812345678")

```

Figure 207: Student window (i)

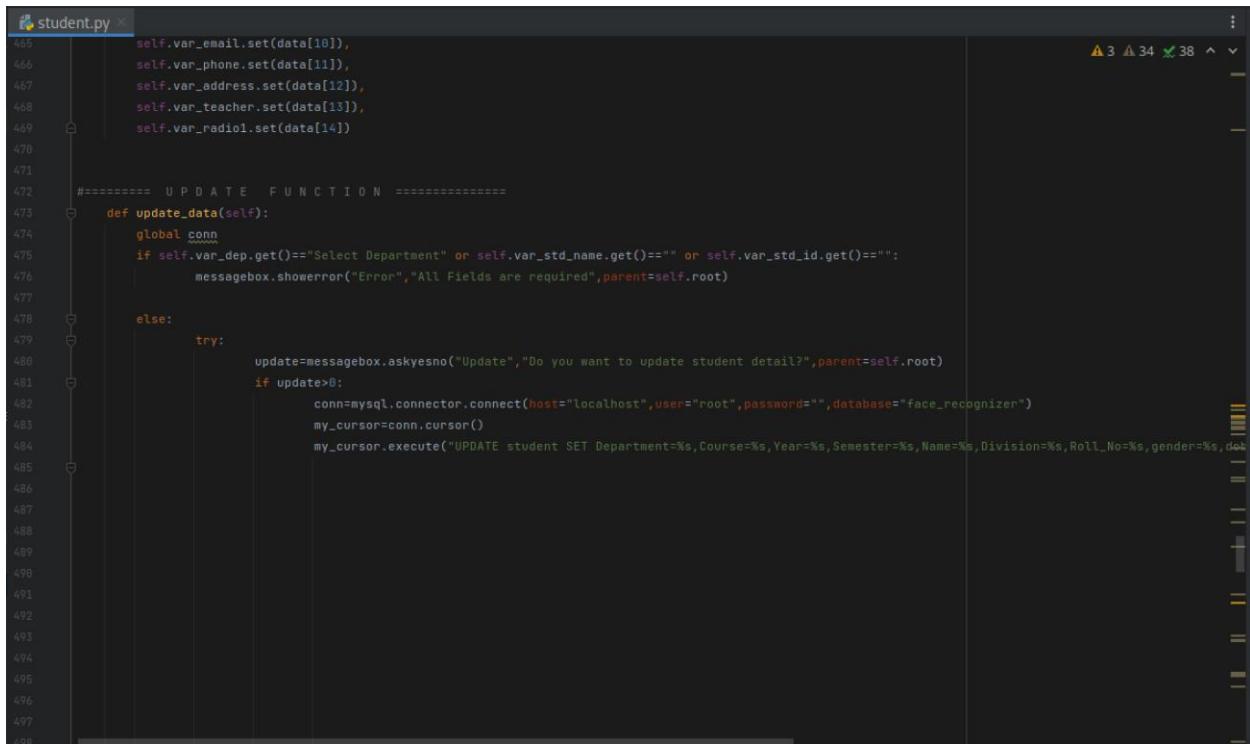
```
student.py
371             return False
372     else:
373         messagebox.showerror("Alert","Invalid entry. Please enter phone (Example: 9812345678")
374     return False
375
376 #ID number validation
377 def checkid(self,id):
378     if len(id) <=10:
379         if id.isdigit():
380             return True
381         if len(str(id))==0:
382             return True
383         else:
384             messagebox.showerror("Invalid","Invalid ID, Please enter ID as integer value(example: 12345")
385         return False
386     else:
387         messagebox.showerror("Invalid","Invalid ID, please enter valid ID")
388
389 #Roll Number validation
390 def checkroll(self,roll):
391     if len(roll) <=6:
392         if roll.isdigit():
393             return True
394         if len(str(roll))==0:
395             return True
396         else:
397             messagebox.showerror("Invalid","Invalid Roll number")
398         return False
399     else:
400         messagebox.showerror("Alert","Invalid roll number,please enter a number ")
401     return False
402
```

Figure 208: Student window (ii)

Figure 209: Student window (iii)

```
student.py
435  #=====Fetch data=====
436  def fetch_data(self):
437      conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
438      my_cursor=conn.cursor()
439      my_cursor.execute("SELECT * FROM student")
440      my_data=my_cursor.fetchall()
441
442      if len(my_data) != 0:
443          self.student_table.delete(*self.student_table.get_children())
444          for i in my_data:
445              self.student_table.insert("",END,values=i)
446          conn.commit()
447
448      conn.close()
449
450  #=====Get Cursor function=====
451  def get_cursor(self,event=""):
452      cursor_focus=self.student_table.focus()
453      table_content=self.student_table.item(cursor_focus)
454      data=table_content["values"]
455
456      self.var_dep.set(data[0]),
457      self.var_course.set(data[1]),
458      self.var_year.set(data[2]),
459      self.var_semester.set(data[3]),
460      self.var_std_id.set(data[4]),
461      self.var_std_name.set(data[5]),
462      self.var_div.set(data[6]),
463      self.var_roll.set(data[7]),
464      self.var_gender.set(data[8]),
465      self.var_dob.set(data[9]),
466      self.var_email.set(data[10]),
467      self.var_phone.set(data[11]),
468      self.var_address.set(data[12]),
```

Figure 210: Student window (iv)

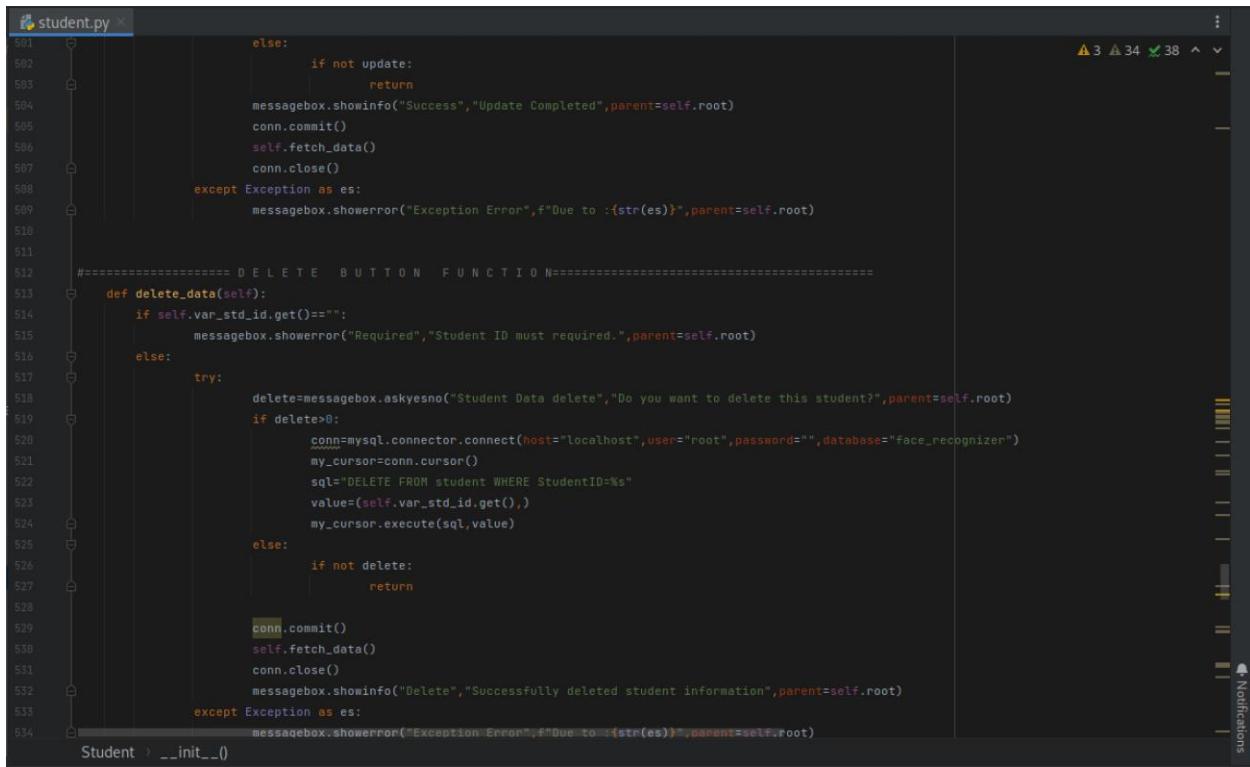


```

465     self.var_email.set(data[10]),
466     self.var_phone.set(data[11]),
467     self.var_address.set(data[12]),
468     self.var_teacher.set(data[13]),
469     self.var_radio1.set(data[14])
470
471
472     ====== U P D A T E   F U N C T I O N ======
473     def update_data(self):
474         global conn
475         if self.var_dep.get()=="Select Department" or self.var_std_name.get()=="" or self.var_std_id.get()=="":
476             messagebox.showerror("Error","All Fields are required",parent=self.root)
477
478         else:
479             try:
480                 update=messagebox.askyesno("Update","Do you want to update student detail?",parent=self.root)
481                 if update>0:
482                     conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
483                     my_cursor=conn.cursor()
484                     my_cursor.execute("UPDATE student SET Department=%s,Course=%s,Year=%s,Semester=%s,Name=%s,Division=%s,Roll_No=%s,gender=%s,dob=%s WHERE StudentID=%s")
485
486
487
488
489
490
491
492
493
494
495
496
497
498

```

Figure 211: Student window (v)



```

501
502         else:
503             if not update:
504                 return
505             messagebox.showinfo("Success","Update Completed",parent=self.root)
506             conn.commit()
507             self.fetch_data()
508             conn.close()
509         except Exception as es:
510             messagebox.showerror("Exception Error",f"Due to :{str(es)}",parent=self.root)
511
512     ====== D E L E T E   B U T T O N   F U N C T I O N ======
513     def delete_data(self):
514         if self.var_std_id.get()=="":
515             messagebox.showerror("Required","Student ID must required.",parent=self.root)
516         else:
517             try:
518                 delete=messagebox.askyesno("Student Data delete","Do you want to delete this student?",parent=self.root)
519                 if delete>0:
520                     conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
521                     my_cursor=conn.cursor()
522                     sql="DELETE FROM student WHERE StudentID=%s"
523                     value=(self.var_std_id.get(),)
524                     my_cursor.execute(sql,value)
525
526                 else:
527                     if not delete:
528                         return
529
530                 conn.commit()
531                 self.fetch_data()
532                 conn.close()
533                 messagebox.showinfo("Delete","Successfully deleted student information",parent=self.root)
534             except Exception as es:
535                 messagebox.showerror("Exception Error",f"Due to :{str(es)}",parent=self.root)

```

Figure 212: Student window (vi)

Figure 213: Student window (vii)

```

student.py
585     conn.commit()
586     self.fetch_data()
587     self.reset_data()
588     conn.close()
589
590     #=====Load predefined data of frontal face from OpenCV=====
591     face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
592
593     #Object detection function
594     def face_cropped(img):
595         #Grayscale convert
596         gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
597         faces=face_classifier.detectMultiScale(gray,1.3,5) #scaling factor=1.3
598                                         #Minimum Neighbor=5
599         for (x,y,w,h) in faces:
600             face_cropped=img[y:y+h+20,x:x+w+20]
601             return face_cropped
602
603     cap=cv2.VideoCapture(-1)
604
605     img_id=0
606     while True:
607         ret,my_frame=cap.read()
608         if face_cropped(my_frame) is not None:
609             img_id+=1
610             face=cv2.resize(face_cropped(my_frame),(450,450))
611             face=cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)
612             file_path="Data/""+user""+str(id)+""+str(img_id)+""".jpg" #File naming with format
613             cv2.imwrite(file_path,face)
614             cv2.putText(face,str(img_id),(50,50),cv2.FONT_HERSHEY_COMPLEX,2,(0,255,0),2)
615             cv2.imshow("Cropped Face",face)
616
617             if cv2.waitKey(1)==13 or int(img_id)==20:
618                 break

```

Figure 214: Student window (viii)

#### 8.6.2.4 Train Window

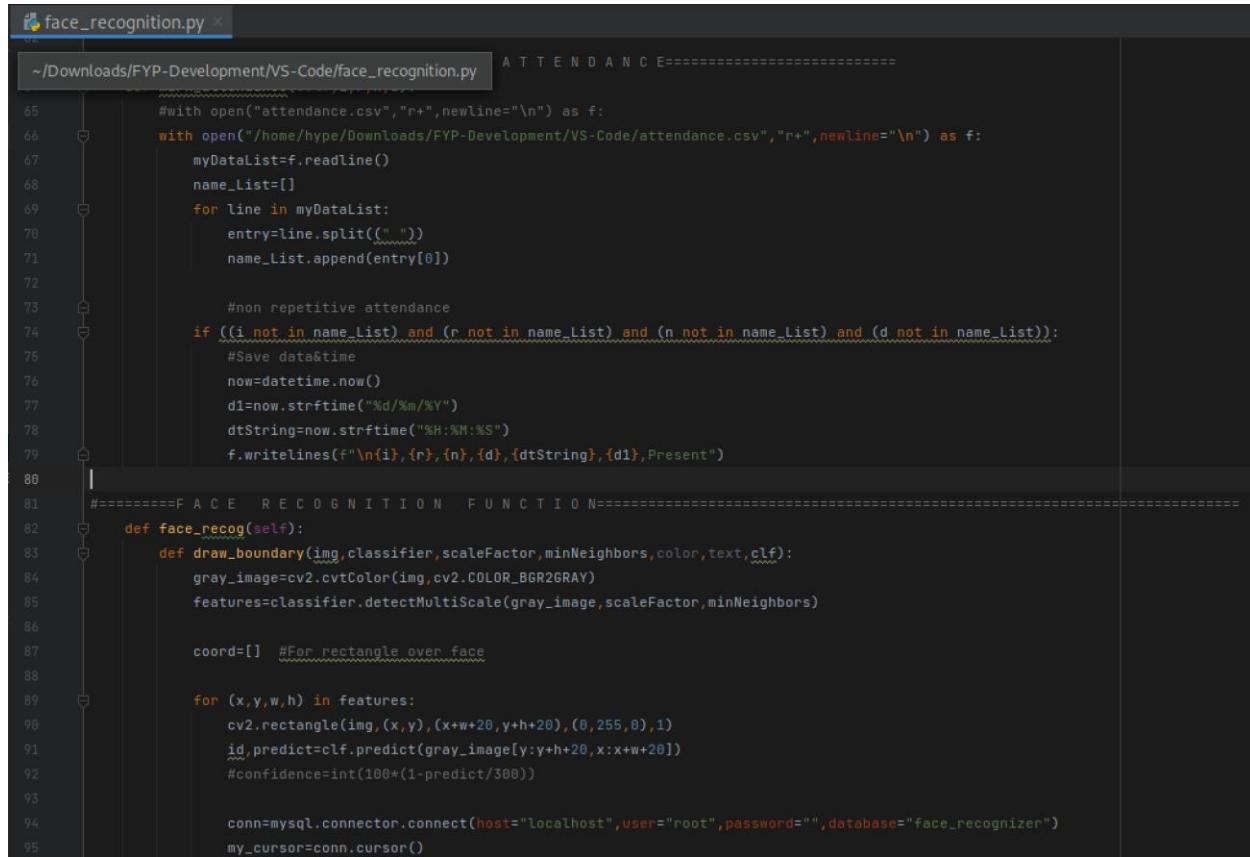
```
    def train_classifier(self):
        data_dir=(r"/home/hype/Downloads/FYP-Development/VS-Code/Data")
        path=[os.path.join(data_dir,file) for file in os.listdir(data_dir)]

        faces=[]
        ids=[]
        for image in path:
            img=Image.open(image).convert('L')      #Gray Scale image
            imageNp=np.array(img,'uint8')  #uint8 is datatype
            id=int(os.path.split(image)[1].split('.')[1])

            faces.append(imageNp)
            ids.append(id)
            cv2.imshow("Training Data",imageNp)
            cv2.waitKey(1)==13
        ids=np.array(ids)
```

Figure 215: Train window

### 8.6.2.5 Recognition Window

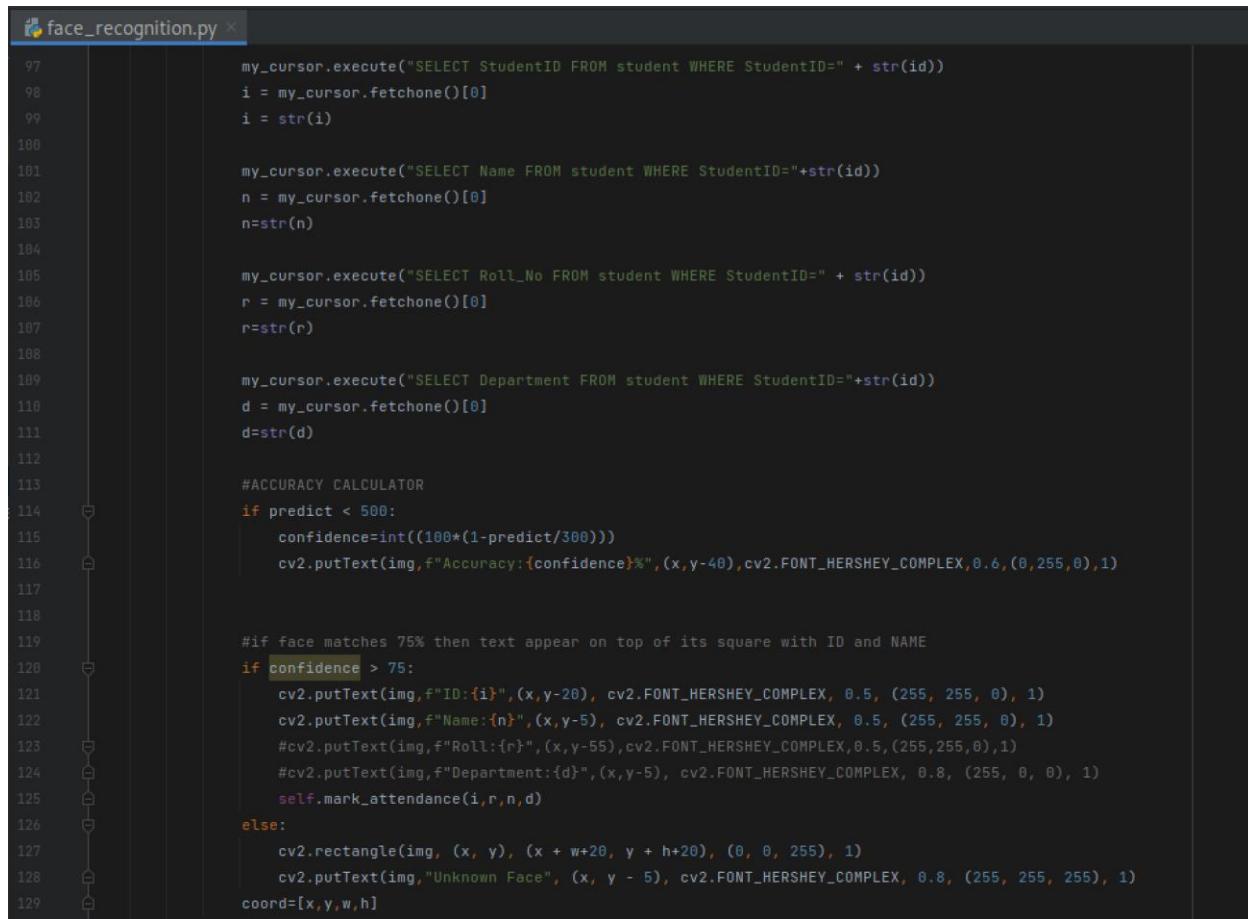


```

face_recognition.py
=====
~/Downloads/FYP-Development/VS-Code/face_recognition.py | A T T E N D A N C E =====
65     #with open("attendance.csv","r+",newline="\n") as f:
66     with open("/home/hype/Downloads/FYP-Development/VS-Code/attendance.csv","r+",newline="\n") as f:
67         myDataList=f.readlines()
68         name_List=[]
69         for line in myDataList:
70             entry=line.split(" ")
71             name_List.append(entry[0])
72
73             #non repetitive attendance
74             if ((i not in name_List) and (r not in name_List) and (n not in name_List) and (d not in name_List)):
75                 #Save data&time
76                 now=datetime.now()
77                 d1=now.strftime("%d/%m/%Y")
78                 dtString=now.strftime("%H:%M:%S")
79                 f.writelines(f"\n{i},{r},{n},{d},{dtString},{d1},Present")
80
81 =====FACE RECOGNITION FUNCTION=====
82     def face_recog(self):
83         def draw_boundary(img,classifier,scaleFactor,minNeighbors,color,text,clf):
84             gray_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
85             features=classifier.detectMultiScale(gray_image,scaleFactor,minNeighbors)
86
87             coord=[] #For rectangle over face
88
89             for (x,y,w,h) in features:
90                 cv2.rectangle(img,(x,y),(x+w+20,y+h+20),(0,255,0),1)
91                 id,predict=clf.predict(gray_image[y:y+h+20,x:x+w+20])
92                 confidence=int(100*(1-predict/300))
93
94                 conn=mysql.connector.connect(host="localhost",user="root",password="",database="face_recognizer")
95                 my_cursor=conn.cursor()

```

Figure 216: Recognition window (i)



```

97         my_cursor.execute("SELECT StudentID FROM student WHERE StudentID=" + str(id))
98         i = my_cursor.fetchone()[0]
99         i = str(i)
100
101        my_cursor.execute("SELECT Name FROM student WHERE StudentID="+str(id))
102        n = my_cursor.fetchone()[0]
103        n=str(n)
104
105        my_cursor.execute("SELECT Roll_No FROM student WHERE StudentID=" + str(id))
106        r = my_cursor.fetchone()[0]
107        r=str(r)
108
109        my_cursor.execute("SELECT Department FROM student WHERE StudentID="+str(id))
110        d = my_cursor.fetchone()[0]
111        d=str(d)
112
113        #ACCURACY CALCULATOR
114        if predict < 500:
115            confidence=int((100*(1-predict/300)))
116            cv2.putText(img,f"Accuracy:{confidence}%",(x,y-40),cv2.FONT_HERSHEY_COMPLEX,0.6,(0,255,0),1)
117
118
119        #if face matches 75% then text appear on top of its square with ID and NAME
120        if confidence > 75:
121            cv2.putText(img,f"ID:{i}",(x,y-20), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 0), 1)
122            cv2.putText(img,f"Name:{n}",(x,y-5), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 0), 1)
123            #cv2.putText(img,f"Roll:{r}",(x,y-55),cv2.FONT_HERSHEY_COMPLEX,0.5,(255,255,0),1)
124            #cv2.putText(img,f"Department:{d}",(x,y-5), cv2.FONT_HERSHEY_COMPLEX, 0.8, (255, 0, 0), 1)
125            self.mark_attendance(i,r,n,d)
126        else:
127            cv2.rectangle(img, (x, y), (x + w+20, y + h+20), (0, 0, 255), 1)
128            cv2.putText(img,"Unknown Face", (x, y - 5), cv2.FONT_HERSHEY_COMPLEX, 0.8, (255, 255, 255), 1)
129            coord=[x,y,w,h]

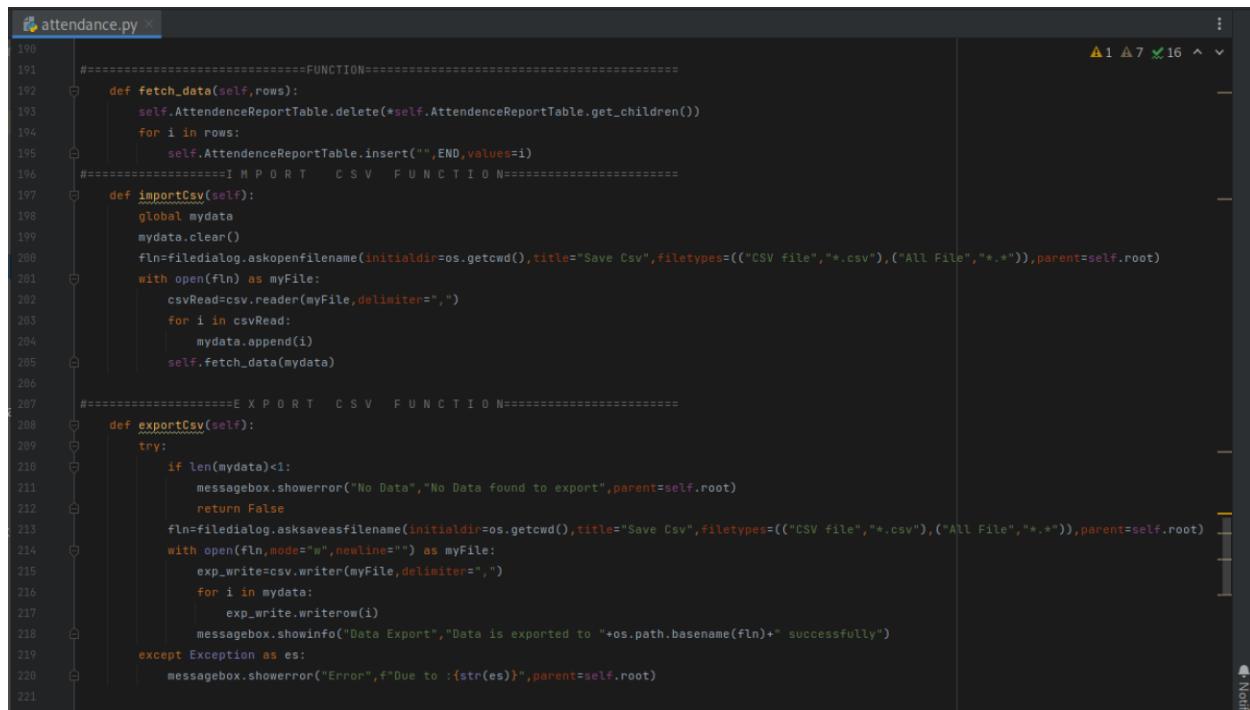
```

Figure 217: Recognition window (ii)

```
 130     return coord
 131
 132 #Image Recognition function
 133 def recognize(img,clf,faceCascade):
 134     coord=draw_boundary(img,faceCascade,1.1,10,(255,25,255),"Face",clf)
 135     return img
 136
 137 faceCascade=cv2.CascadeClassifier("/home/hype/Downloads/FYP-Development/VS-Code/haarcascade_frontalface_default.xml")
 138 clf=cv2.face.LBPHFaceRecognizer_create()
 139 clf.read("/home/hype/Downloads/FYP-Development/VS-Code/Classifier.xml")
 140
 141 #Capture Video with index (-10,-9,-8,-7,-6,-5,-4,-3,-2,-1) to open camera// tested in camera module
 142 video_cap=cv2.VideoCapture(-1)
 143
 144 while True:
 145     ret,img=video_cap.read()
 146     img=recognize(img,clf,faceCascade)
 147     cv2.imshow("Welcome to FRAS",img)
 148     #cv2.waitKey(1)
 149     if cv2.waitKey(1)==13:
 150         break
 151     video_cap.release()
 152     cv2.destroyAllWindows()
 153
 154 #df_state=pd.read_csv(r"/home/hype/Downloads/FYP-Development/VS-Code/attendance.csv")
 155 #DF_RM_DUP=df_state.drop_duplicate(keep=False)
 156 #DF_RM_DUP.to_csv("test.csv",index=False)
 157
 158
 159 >>> if __name__ == "__main__":
 160     root=Tk()
 161     obj=face_recognition(root)
 162     root.mainloop()
```

Figure 218: Recognition window (iii)

### 8.6.2.6 Attendance Window

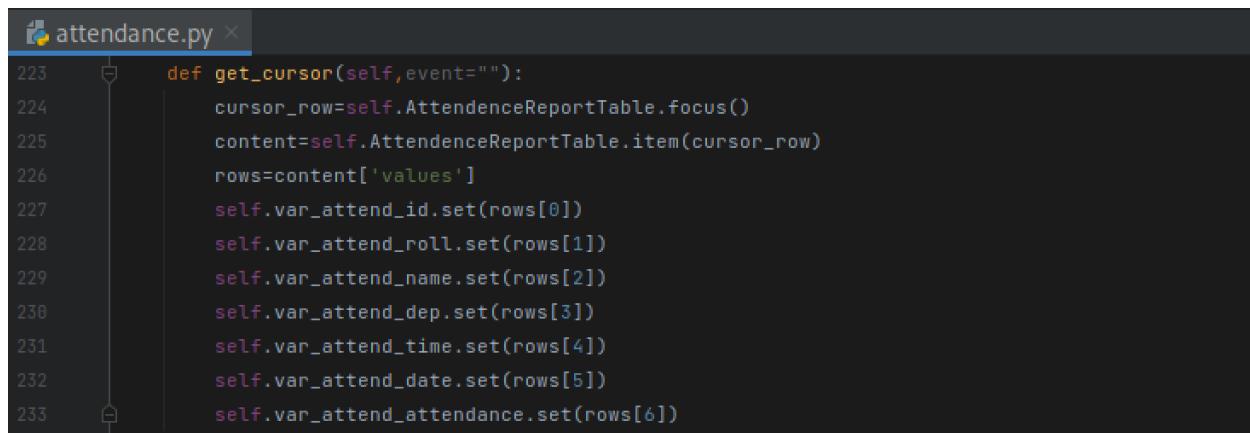


```

190
191     #=====FUNCTION=====
192     def fetch_data(self,rows):
193         self.AttendanceReportTable.delete(*self.AttendanceReportTable.get_children())
194         for i in rows:
195             self.AttendanceReportTable.insert("",END,values=i)
196     #=====IMP OR T CSV F U N C T I O N =====
197     def importCsv(self):
198         global mydata
199         mydata.clear()
200         fln=filedialog.askopenfilename(initialdir=os.getcwd(),title="Save Csv",filetypes=(("CSV file","*.csv"),("All File","*.*")),parent=self.root)
201         with open(fln) as myFile:
202             csvRead=csv.reader(myFile,delimiter=",")
203             for i in csvRead:
204                 mydata.append(i)
205             self.fetch_data(mydata)
206
207     #=====E X P O R T CSV F U N C T I O N =====
208     def exportCsv(self):
209         try:
210             if len(mydata)<1:
211                 messagebox.showerror("No Data","No Data found to export",parent=self.root)
212                 return False
213             fln=filedialog.asksaveasfilename(initialdir=os.getcwd(),title="Save Csv",filetypes=(("CSV file","*.csv"),("All File","*.*")),parent=self.root)
214             with open(fln,mode="w",newline="") as myFile:
215                 exp_write=csv.writer(myFile,delimiter=",")
216                 for i in mydata:
217                     exp_write.writerow(i)
218                 messagebox.showinfo("Data Export","Data is exported to "+os.path.basename(fln)+" successfully")
219         except Exception as es:
220             messagebox.showerror("Error",f"Due to :{str(es)}",parent=self.root)
221

```

Figure 219: Attendance window (i)



```

223
224     def get_cursor(self,event=""):
225         cursor_row=self.AttendanceReportTable.focus()
226         content=self.AttendanceReportTable.item(cursor_row)
227         rows=content['values']
228         self.var_attend_id.set(rows[0])
229         self.var_attend_roll.set(rows[1])
230         self.var_attend_name.set(rows[2])
231         self.var_attend_dep.set(rows[3])
232         self.var_attend_time.set(rows[4])
233         self.var_attend_date.set(rows[5])
234         self.var_attend_attendance.set(rows[6])

```

Figure 220: Attendance window(ii)

```
    def reset_data(self):
        self.var_attend_id.set("")
        self.var_attend_roll.set("")
        self.var_attend_dep.set("")
        self.var_attend_name.set("")
        self.var_attend_time.set("")
        self.var_attend_date.set("")
        self.var_attend_attendance.set("")

    if __name__ == "__main__":
        root=Tk()
        obj=attendance(root)
        root.mainloop()
```

Figure 221: Attendance window (iii)

## 8.7 Appendix D: Designs

### 8.7.1 Gantt Chart

#### 8.7.1.1 Initial Gantt Chart

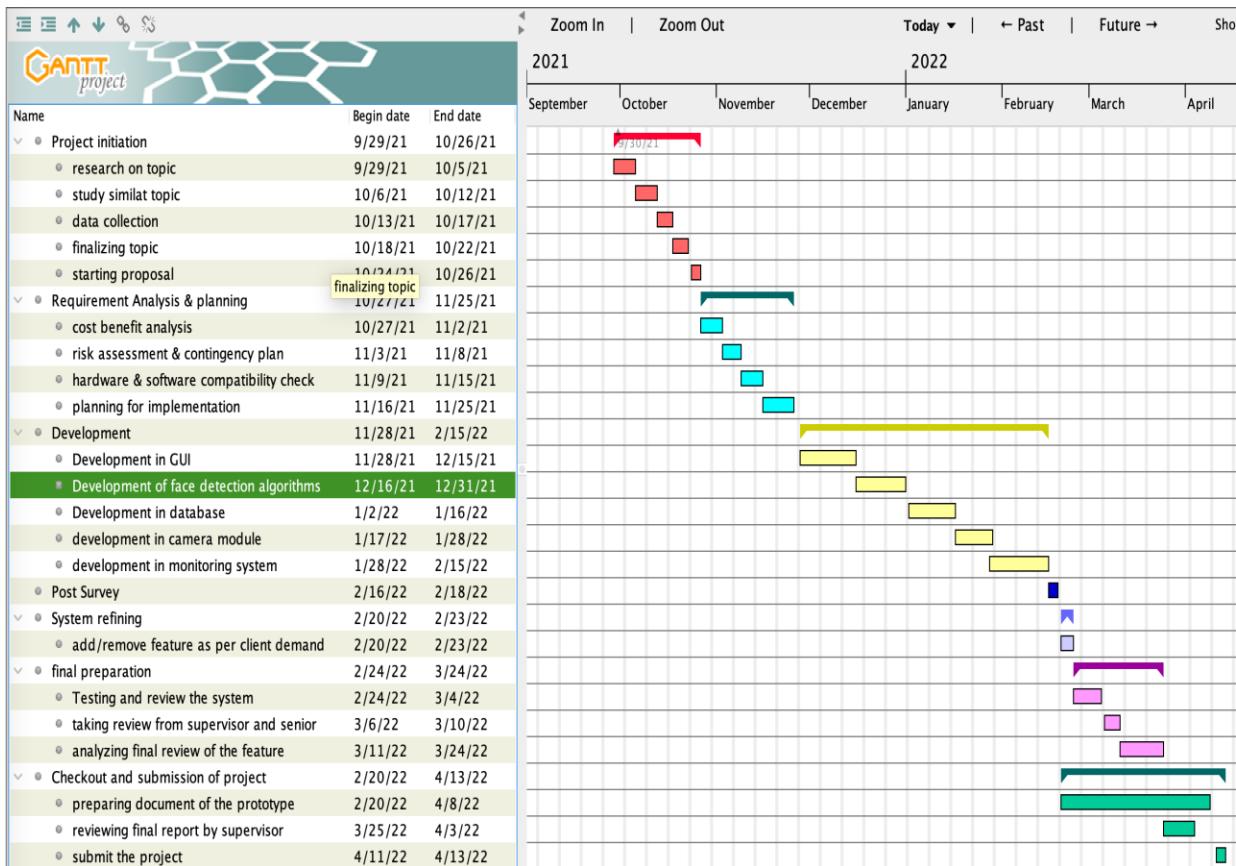


Figure 222: Initial Gantt Chart

### 8.7.1.2 Final Gantt Chart

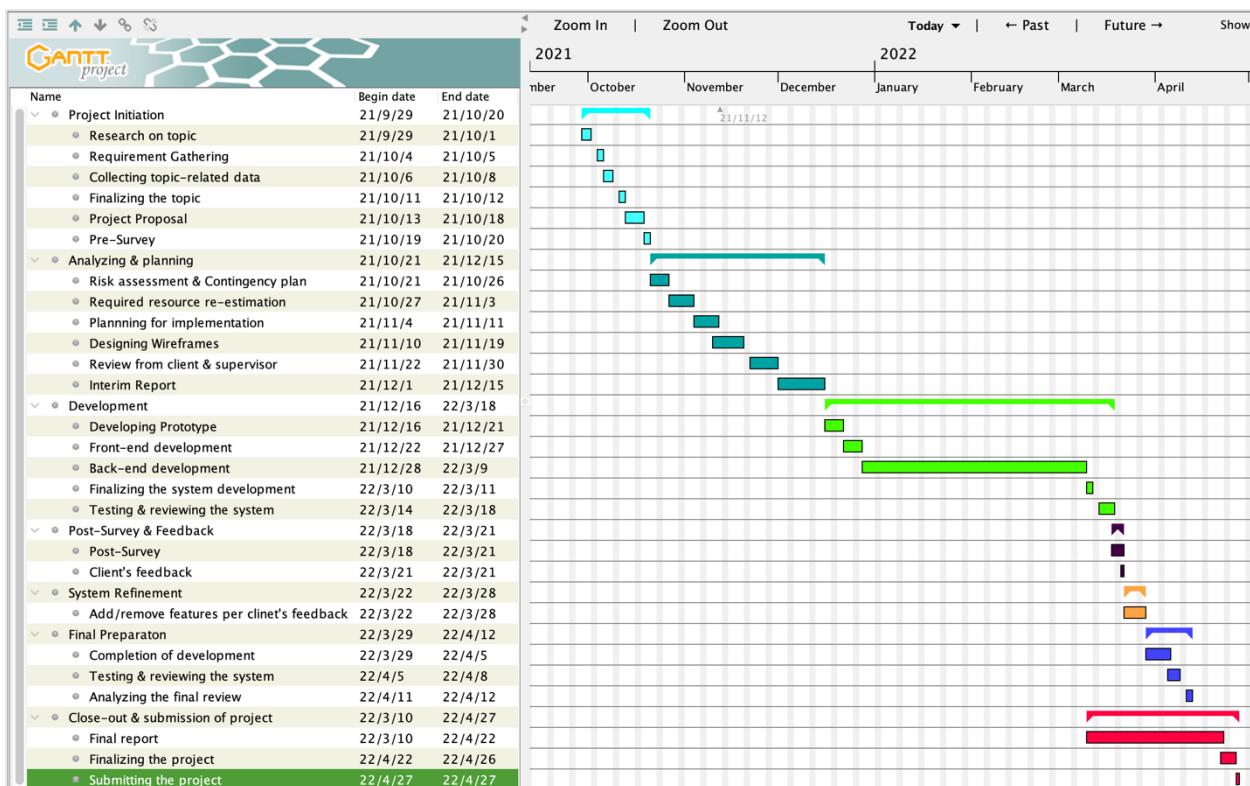


Figure 223: Final Gantt Chart

### 8.7.2 Work Breakdown Structure

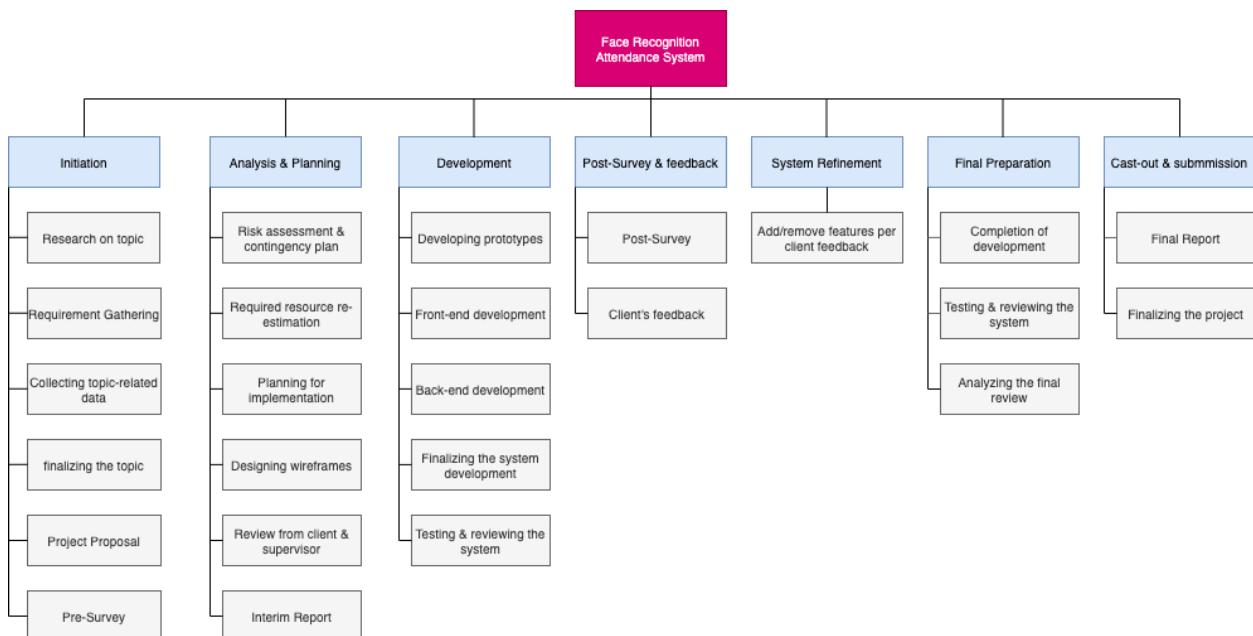


Figure 224: Work Breakdown Structure

**8.7.3 Algorithms****8.7.3.1 Admin/Teacher login Algorithm**

Step 1: Start

Step 2: Go to login window

Step 3: Do you have account?

Step 3.1: If yes, Do login to account

Step 3.1.1: Go to Home Page

Step 3.1.2: Manage Student Detail, Image Process and manipulate

Step 3.1.3: Export the attendance detail

Step 3.1.4: Exit

Step 4: If no account, do register and go to step 3.1

Step 5: Stop

**8.7.3.2 Recognition and attendance algorithm**

Step 1: Start

Step 2: Image stored in system's database

Step 3: Recognition Process start

Step 4: Camera launches and start capture images

Step 5: Compare with database

Step 5.1: If Match

Step 5.2: Present to the student

Step 5.3: Student ID and Name display in screen

Step 5.4: CSV file generate with attendance

Step 5.5: Exit, go to step 4

Step 6: No Match, go to step 4

Step 7: Stop

#### 8.7.4 Activity Diagram

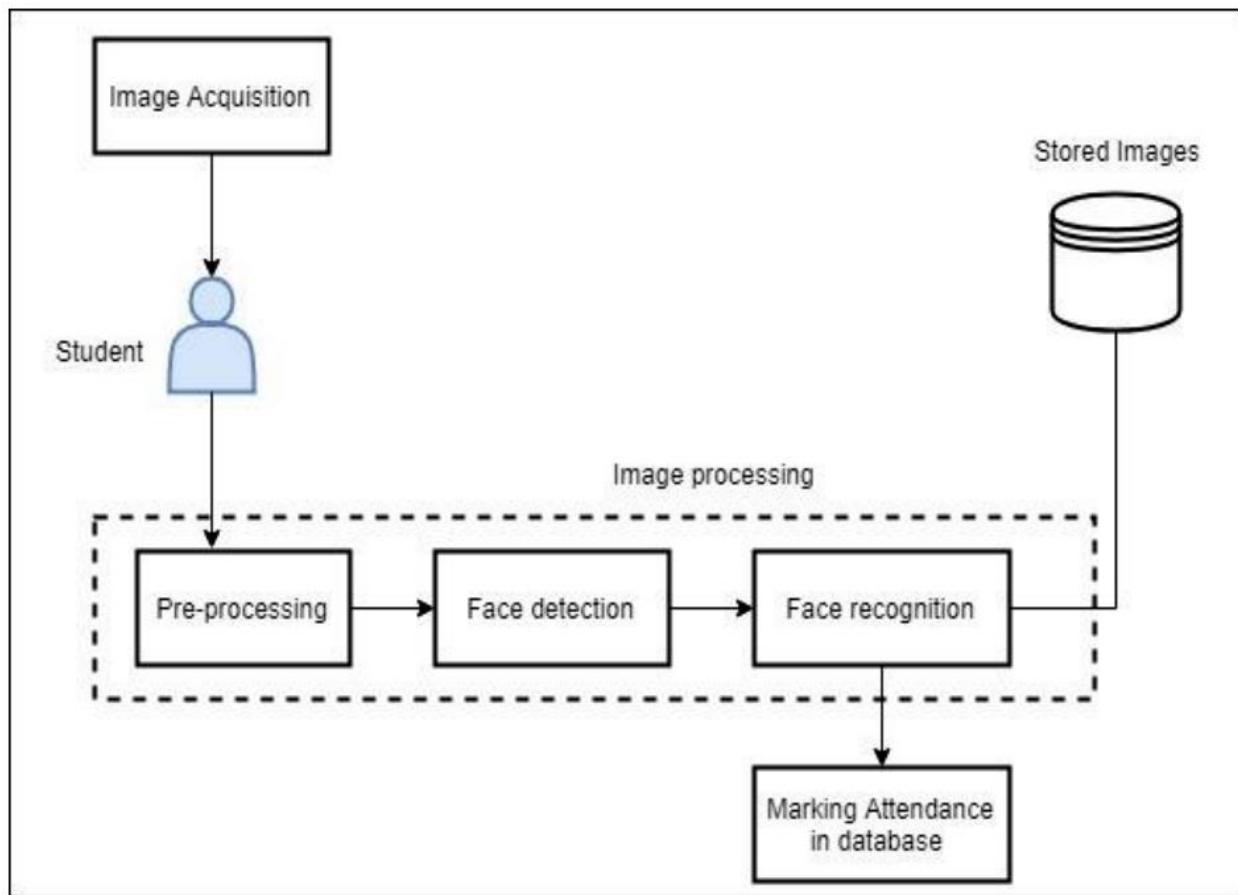


Figure 225: Activity diagram

### 8.7.5 Data Flow Diagram

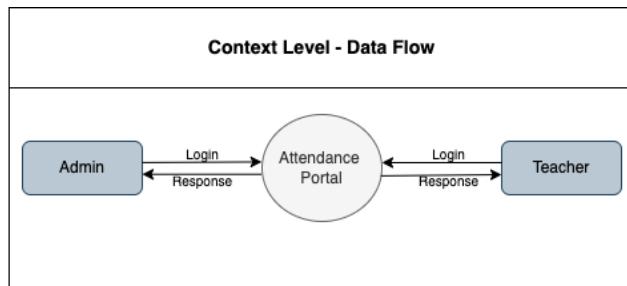


Figure 226: Data Flow Diagram - Context Level

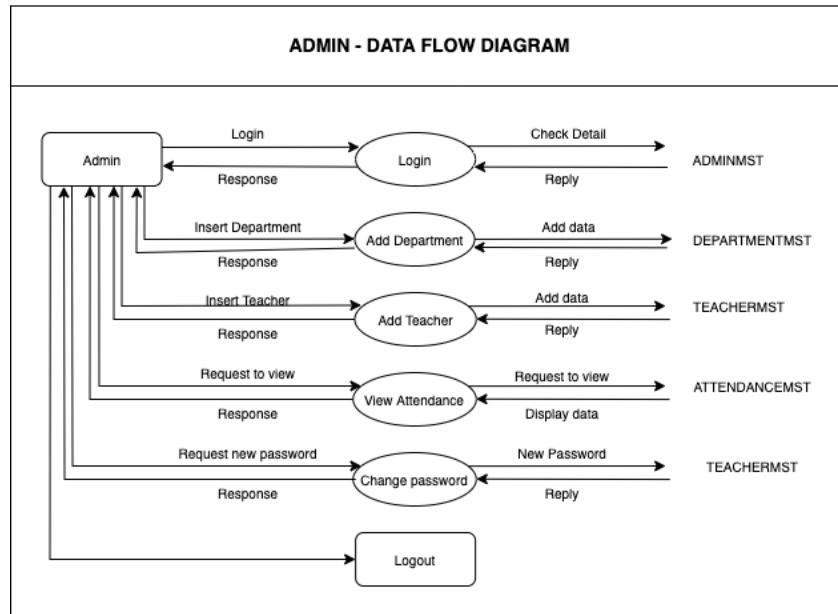


Figure 227: Data flow diagram - Admin

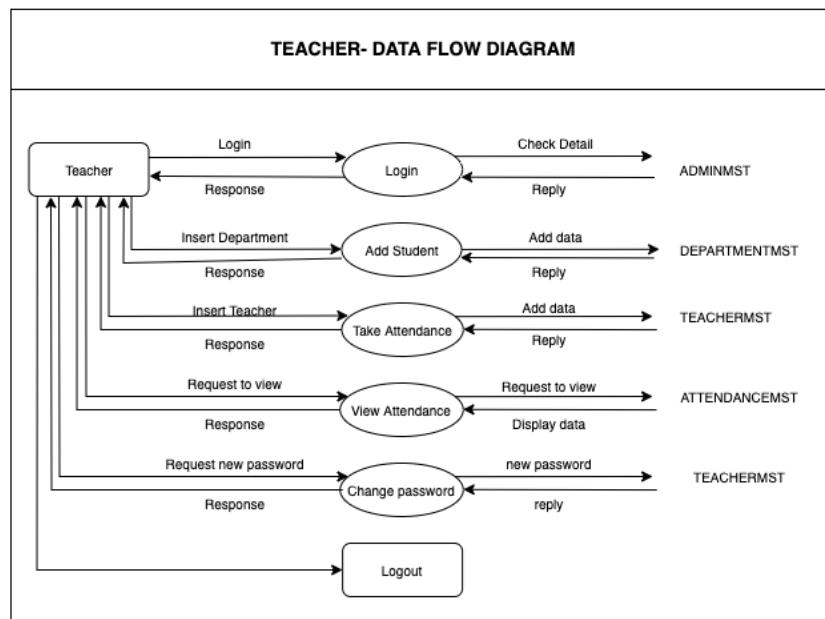


Figure 228: Data Flow Diagram – Teacher