**Student Enrollment System**

* Prepared by:
  + Hozan Ageed Farho
  + Ismail Nouri Abozed
  + Usama Farhad Muhammad
* Supervised by:
  + Dr. Haidar Haddad

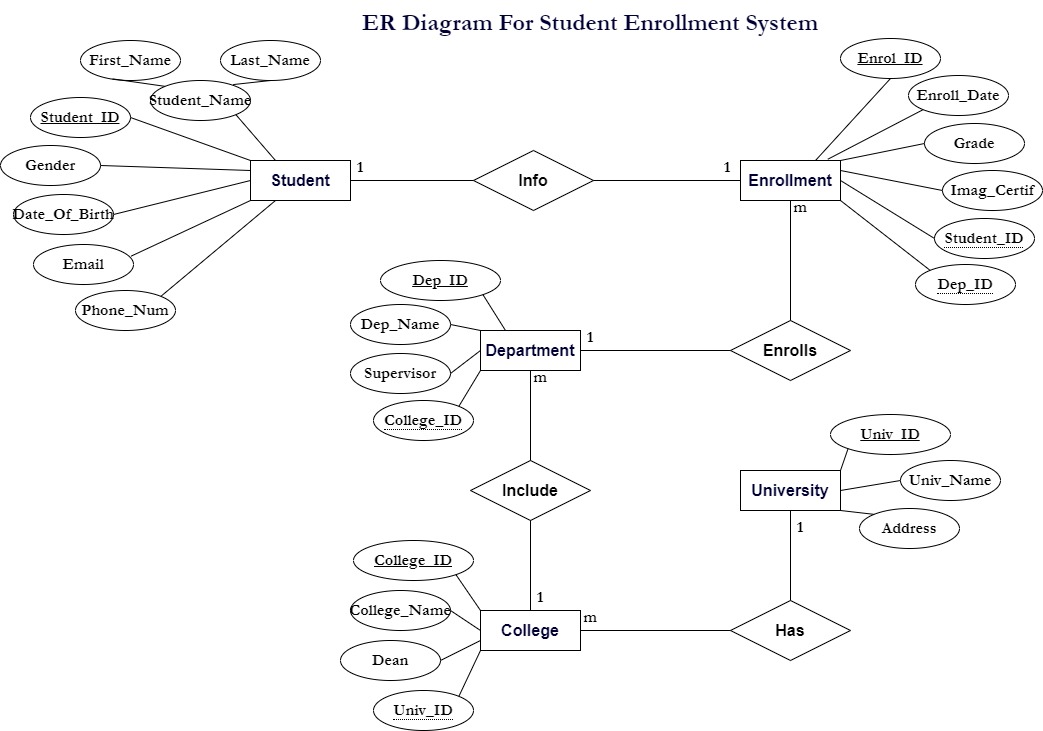
**College of Science /**

**Computer Science & IT Department.**

**2020-2021**



**ER Diagram**



**Relational Model:**

**Student** (Student\_ID, First\_Name, Last\_Name, Gender, Date\_Of\_Birth,

Email, Phone\_Num)

**Enrollment** (Enrol\_ID, Enroll\_Date, Grade, Imag\_Certif, Student\_ID, Dep\_ID)

**Department** (Dep\_ID, Dep\_Name, Supervisor, College\_ID)

**College** (College\_ID, College\_Name, Dean, Univ\_ID)

**University** (Univ\_ID, Univ\_Name, Adress)

**Relational Model Relation:**

**University**

|  |  |  |
| --- | --- | --- |
| Univ\_ID | Univ\_Name | Adress |
| 1 | Salahaddin | Erbil-karkuk-street |
| 2 | Sulaimani | Sulaimani-street |
| 3 | Duhok | Duho-Zakho-street |

**College**

|  |  |  |  |
| --- | --- | --- | --- |
| College\_ID | College\_Name | Dean | Univ\_ID |
| 11 | Science | Professor Hauresh | 1 |
| 12 | Engineering | Professor Hamdin | 1 |
| 13 | Language | Professor Ahmad | 1 |
| 14 | Engineering | Professor Ramadan | 2 |
| 15 | Medicine | Professor Ali | 2 |
| 16 | Language | Professor Sarkheel | 3 |
| 17 | Science | Professor Shekhmus | 3 |
| 18 | Engineering | Professor Zana | 3 |
| 19 | Medicine | Professor Aland | 3 |

**Department**

|  |  |  |  |
| --- | --- | --- | --- |
| Dep\_ID | Dep\_Name | Supervisor | College\_ID |
| 123 | CS & IT | Dr Dler | 11 |
| 124 | Math | Dr Maria | 11 |
| 125 | Physic | Dr Nushi | 11 |
| 126 | Geology | Dr Mohammed | 11 |
| 127 | Biology | Dr Jose | 11 |
| 128 | chemic | Dr Wei | 11 |
| 129 | Software Engineering | Dr Salar | 12 |
| 130 | Mechanical & Mechatronics | Dr Yan | 12 |
| 131 | Architecture | Dr John | 12 |
| 132 | Civil | Dr David | 12 |
| 133 | Electrical | Dr Li | 12 |
| 134 | Kurdish | Dr Eanoo | 13 |
| 135 | Arabic | Dr Abdul | 13 |
| 136 | English | Dr Ana | 13 |
| 137 | French | Dr Ying | 13 |
| 138 | Germany | Dr Michael | 13 |
| 139 | Turkish | Dr Juan | 13 |
| 140 | Software Engineering | Dr Anna | 14 |
| 141 | Mechanical & Mechatronics | Dr Mary | 14 |
| 142 | Architecture | Dr Jean | 14 |
| 143 | Civil | Dr Robert | 14 |
| 144 | Electrical | Dr Daniel | 14 |
| 145 | General Medicine | Dr Luis | 15 |
| 146 | Dentistry | Dr Carlos | 15 |
| 147 | Pharmacy | Dr James | 15 |
| 148 | Ophthalmology | Dr Antonio | 15 |
| 149 | Kurdish | Dr Joseph | 16 |
| 150 | Arabic | Dr Hui | 16 |
| 151 | English | Dr Elena | 16 |
| 152 | French | Dr Francisco | 16 |
| 153 | Germany | Dr Hong | 16 |
| 154 | Turkish | Dr Marie | 16 |
| 155 | CS & IT | Dr Min | 17 |
| 156 | Math | Dr Lei | 17 |
| 157 | Physic | Dr Yu | 17 |
| 158 | Geology | Dr Ibrahim | 17 |
| 159 | Biology | Dr Peter | 17 |
| 160 | chemic | Dr Fatima | 17 |
| 161 | Software Engineering | Dr Aleksandr | 18 |
| 162 | Mechanical & Mechatronics | Dr Richard | 18 |
| 163 | Architecture | Dr Xin | 18 |
| 164 | Civil | Dr Bin | 18 |
| 165 | Electrical | Dr Paul | 18 |
| 166 | General Medicine | Dr Ping | 19 |
| 167 | Dentistry | Dr Lin | 19 |
| 168 | Pharmacy | Dr Olga | 19 |
| 169 | Ophthalmology | Dr Sri | 19 |

**Student**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student\_ID | First\_Name | Last\_Name | Gender | Date\_Of\_Birth | Email | Phone\_Number |
| 112200 | James | John | Male | 1999-07-01 | James@gmail.com | +9647826030525 |
| 112201 | Mary | Patricia | Female | 2000-07-02 | Mary@gmail.com | +9647835459526 |
| 112202 | Robert | Michael | Male | 2000-07-03 | Robert@gmail.com | +9647507788527 |
| 112203 | Jennifer | Linda | Female | 2000-07-04 | Jennifer@gmail.com | +9647835459528 |
| 112204 | William | David | Male | 1999-07-05 | William@gmail.com | +9647826030529 |
| 112205 | Elizabeth | Barbara | Female | 2000-07-06 | Elizabeth@gmail.com | +9647835459530 |
| 112206 | Richard | Joseph | Male | 2000-07-07 | Richard@gmail.com | +9647507788531 |
| 112207 | Susan | Jessica | Female | 2000-07-08 | Susan@gmail.com | +9647835459532 |
| 112208 | Thomas | Charles | Male | 1999-07-09 | Thomas@gmail.com | +9647826030533 |
| 112209 | Sarah | Karen | Female | 2000-07-10 | Sarah@gmail.com | +9647835459534 |
| 112210 | Christopher | Daniel | Male | 2000-07-11 | Christopher@gmail.com | +9647507788535 |
| 112211 | Nancy | Lisa | Female | 2000-07-12 | Nancy@gmail.com | +9647835459536 |
| 112212 | Matthew | Anthony | Male | 1999-07-13 | Matthew@gmail.com | +9647826030537 |
| 112213 | Margaret | Betty | Female | 2000-07-14 | Margaret@gmail.com | +9647835459538 |
| 112214 | Donald | Mark | Male | 2000-07-15 | Donald@gmail.com | +9647507788539 |
| 112215 | Sandra | Ashley | Female | 2000-07-16 | Sandra@gmail.com | +9647835459540 |
| 112216 | Paul | Steven | Male | 1999-07-17 | Paul@gmail.com | +9647826030541 |
| 112217 | Dorothy | Kimberly | Female | 2000-07-18 | Dorothy@gmail.com | +9647835459542 |
| 112218 | Andrew | Kenneth | Male | 2000-07-19 | Andrew@gmail.com | +9647507788543 |
| 112219 | Emily | Donna | Female | 2000-07-20 | Emily@gmail.com | +9647835459544 |
| 112220 | Joshua | Kevin | Male | 1999-07-21 | Joshua@gmail.com | +9647826030545 |
| 112221 | Michelle | Carol | Female | 2000-07-22 | Michelle@gmail.com | +9647835459546 |
| 112222 | Brian | George | Male | 2000-07-23 | Brian@gmail.com | +9647507788547 |
| 112223 | Amanda | Melissa | Female | 2000-07-24 | Amanda@gmail.com | +9647835459548 |
| 112224 | Edward | Ronald | Male | 1999-07-25 | Edward@gmail.com | +9647826030549 |
| 112225 | Deborah | Stephanie | Female | 2000-08-01 | Deborah@gmail.com | +9647835459550 |
| 112226 | Timothy | Jason | Male | 2000-08-02 | Timothy@gmail.com | +9647507788551 |
| 112227 | Rebecca | Laura | Female | 2000-08-03 | Rebecca@gmail.com | +9647835459552 |
| 112228 | Jeffrey | Ryan | Male | 1999-08-04 | Jeffrey@gmail.com | +9647826030553 |
| 112229 | Sharon | Cynthia | Female | 2000-08-05 | Sharon@gmail.com | +9647835459554 |
| 112230 | Jacob | Gary | Male | 2000-08-06 | Jacob@gmail.com | +9647507788555 |
| 112231 | Kathleen | Amy | Female | 2000-08-07 | Kathleen@gmail.com | +9647835459556 |
| 112232 | Nicholas | Eric | Male | 1999-08-08 | Nicholas@gmail.com | +9647826030557 |
| 112233 | Shirley | Angela | Female | 2000-08-09 | Shirley@gmail.com | +9647835459558 |
| 112234 | Jonathan | Stephen | Male | 2000-08-10 | Jonathan@gmail.com | +9647507788559 |
| 112235 | Helen | Anna | Female | 2000-08-11 | Helen@gmail.com | +9647835459560 |
| 112236 | Larry | Justin | Male | 1999-08-12 | Larry@gmail.com | +9647826030561 |
| 112237 | Brenda | Pamela | Female | 2000-08-13 | Brenda@gmail.com | +9647835459562 |
| 112238 | Scott | Brandon | Male | 2000-08-14 | Scott@gmail.com | +9647507788563 |
| 112239 | Nicole | Samantha | Female | 2000-08-15 | Nicole@gmail.com | +9647835459564 |
| 112240 | Benjamin | Samuel | Male | 1999-08-16 | Benjamin@gmail.com | +9647826030565 |
| 112241 | Katherine | Emma | Female | 2000-08-17 | Katherine@gmail.com | +9647835459566 |
| 112242 | Frank | Gregory | Male | 2000-08-18 | Frank@gmail.com | +9647507788567 |
| 112243 | Ruth | Christine | Female | 2000-08-19 | Ruth@gmail.com | +9647835459568 |
| 112244 | Raymond | Alexander | Male | 1999-08-20 | Raymond@gmail.com | +9647826030569 |
| 112245 | Catherine | Debra | Female | 2000-08-21 | Catherine@gmail.com | +9647835459570 |
| 112246 | Patrick | Jack | Male | 2000-08-22 | Patrick@gmail.com | +9647507788571 |
| 112247 | Rachel | Carolyn | Female | 2000-08-23 | Rachel@gmail.com | +9647835459572 |
| 112248 | Dennis | Jerry | Male | 1999-08-24 | Dennis@gmail.com | +9647826030573 |
| 112249 | Janet | Virginia | Female | 2000-08-25 | Janet@gmail.com | +9647835459574 |
| 112250 | Tyler | Aaron | Male | 2000-09-01 | Tyler@gmail.com | +9647507788575 |
| 112251 | Maria | Heather | Female | 2000-09-02 | Maria@gmail.com | +9647835459576 |
| 112252 | Jose | Henry | Male | 1999-09-03 | Jose@gmail.com | +9647826030577 |
| 112253 | Diane | Julie | Female | 2000-09-04 | Diane@gmail.com | +9647835459578 |
| 112254 | Adam | Douglas | Male | 2000-09-05 | Adam@gmail.com | +9647507788579 |
| 112255 | Joyce | Victoria | Female | 2000-09-06 | Joyce@gmail.com | +9647835459580 |
| 112256 | Nathan | Peter | Male | 1999-09-07 | Nathan@gmail.com | +9647826030581 |
| 112257 | Kelly | Christina | Female | 2000-09-08 | Kelly@gmail.com | +9647835459582 |
| 112258 | Zachary | Kyle | Male | 2000-09-09 | Zachary@gmail.com | +9647507788583 |
| 112259 | Lauren | Joan | Female | 2000-09-10 | Lauren@gmail.com | +9647835459584 |
| 112260 | Walter | Harold | Male | 1999-09-11 | Walter@gmail.com | +9647826030585 |
| 112261 | Evelyn | Olivia | Female | 2000-09-12 | Evelyn@gmail.com | +9647835459586 |
| 112262 | Jeremy | Ethan | Male | 2000-09-13 | Jeremy@gmail.com | +9647507788587 |
| 112263 | Judith | Megan | Female | 2000-09-14 | Judith@gmail.com | +9647835459588 |
| 112264 | Carl | Keith | Male | 1999-09-15 | Carl@gmail.com | +9647826030589 |
| 112265 | Cheryl | Martha | Female | 2000-09-16 | Cheryl@gmail.com | +9647835459590 |
| 112266 | Roger | Gerald | Male | 2000-09-17 | Roger@gmail.com | +9647507788591 |
| 112267 | Andrea | Frances | Female | 2000-09-18 | Andrea@gmail.com | +9647835459592 |
| 112268 | Christian | Terry | Male | 1999-09-19 | Christian@gmail.com | +9647826030593 |
| 112269 | Hannah | Jacqueline | Female | 2000-09-20 | Hannah@gmail.com | +9647835459594 |
| 112270 | Sean | Arthur | Male | 2000-09-21 | Sean@gmail.com | +9647507788595 |
| 112271 | Ann | Gloria | Female | 2000-09-22 | Ann@gmail.com | +9647835459596 |
| 112272 | Austin | Noah | Male | 1999-09-23 | Austin@gmail.com | +9647826030597 |
| 112273 | Jean | Kathryn | Female | 2000-09-24 | Jean@gmail.com | +9647835459598 |
| 112274 | Lawrence | Jesse | Male | 2000-09-25 | Lawrence@gmail.com | +9647507788599 |
| 112275 | Alice | Teresa | Female | 2000-10-01 | Alice@gmail.com | +9647835459600 |
| 112276 | Joe | Bryan | Male | 1999-10-02 | Joe@gmail.com | +9647826030601 |
| 112277 | Sara | Janice | Female | 2000-10-03 | Sara@gmail.com | +9647835459602 |
| 112278 | Billy | Jordan | Male | 2000-10-04 | Billy@gmail.com | +9647507788603 |
| 112279 | Doris | Madison | Female | 2000-10-05 | Doris@gmail.com | +9647835459604 |
| 112280 | Albert | Dylan | Male | 1999-10-06 | Albert@gmail.com | +9647826030605 |
| 112281 | Julia | Grace | Female | 2000-10-07 | Julia@gmail.com | +9647835459607 |
| 112282 | Bruce | Willie | Male | 2000-10-08 | Bruce@gmail.com | +9647507788608 |
| 112283 | Judy | Abigail | Female | 2000-10-09 | Judy@gmail.com | +9647835459609 |
| 112284 | Gabriel | Alan | Male | 1999-10-10 | Gabriel@gmail.com | +9647826030610 |
| 112285 | Marie | Denise | Female | 2000-10-11 | Marie@gmail.com | +9647835459611 |
| 112286 | Juan | Logan | Male | 2000-10-12 | Juan@gmail.com | +9647507788612 |
| 112287 | Beverly | Amber | Female | 2000-10-13 | Beverly@gmail.com | +9647835459613 |
| 112288 | Wayne | Ralph | Male | 1999-10-14 | Wayne@gmail.com | +9647826030614 |
| 112289 | Theresa | Marilyn | Female | 2000-10-15 | Theresa@gmail.com | +9647835459615 |
| 112290 | Roy | Eugene | Male | 2000-10-16 | Roy@gmail.com | +9647507788616 |
| 112291 | Danielle | Diana | Female | 2000-10-17 | Danielle@gmail.com | +9647835459617 |
| 112292 | Randy | Vincent | Male | 1999-10-18 | Randy@gmail.com | +9647826030618 |
| 112293 | Brittany | Natalie | Female | 2000-10-19 | Brittany@gmail.com | +9647835459619 |
| 112294 | Russell | Louis | Male | 2000-10-20 | Russell@gmail.com | +9647507788620 |
| 112295 | Sophia | Rose | Female | 2000-10-21 | Sophia@gmail.com | +9647835459621 |
| 112296 | Philip | Bobby | Male | 1999-10-22 | Philip@gmail.com | +9647826030622 |
| 112297 | Isabella | Alexis | Female | 2000-10-23 | Isabella@gmail.com | +9647835459623 |
| 112298 | Kayla | Charlotte | Male | 2000-10-24 | Kayla@gmail.com | +9647507788624 |
| 112299 | shereen | sulaiman | Female | 2000-10-25 | shirin2021@gmail.com | +9647835459625 |

**Enrollment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Enrol\_ID | Enrollment\_Date | Grade | Image\_Certificate | Student\_ID | Dep\_ID |
| 998800 | 2016-08-01 | 65.2 | images (1).jpg | 112200 | 123 |
| 998801 | 2017-09-01 | 70.9 | images (2).jpg | 112201 | 123 |
| 998802 | 2018-10-01 | 80.7 | images (3).jpg | 112202 | 124 |
| 998803 | 2019-11-01 | 90.4 | images (4).jpg | 112203 | 124 |
| 998804 | 2020-12-01 | 92.6 | images (5).jpg | 112204 | 125 |
| 998805 | 2016-08-02 | 86.7 | images (6).jpg | 112205 | 125 |
| 998806 | 2017-09-02 | 80.2 | images (7).jpg | 112206 | 126 |
| 998807 | 2018-10-02 | 60.5 | images (8).jpg | 112207 | 126 |
| 998808 | 2019-11-02 | 50.5 | images (9).jpg | 112208 | 127 |
| 998809 | 2020-12-02 | 55.5 | images (10).jpg | 112209 | 127 |
| 998810 | 2016-08-03 | 66.7 | images (11).jpg | 112210 | 128 |
| 998811 | 2017-09-03 | 77.2 | images (12).jpg | 112211 | 128 |
| 998812 | 2018-10-03 | 90.5 | images (13).jpg | 112212 | 129 |
| 998813 | 2019-11-03 | 88.5 | images (14).jpg | 112213 | 129 |
| 998814 | 2020-12-03 | 99 | images (15).jpg | 112214 | 130 |
| 998815 | 2016-08-04 | 81.4 | images (16).jpg | 112215 | 130 |
| 998816 | 2017-09-04 | 74.3 | images (17).jpg | 112216 | 131 |
| 998817 | 2018-10-04 | 91.8 | images (18).jpg | 112217 | 131 |
| 998818 | 2019-11-04 | 92.8 | images (19).jpg | 112218 | 132 |
| 998819 | 2020-12-04 | 93.8 | images (20).jpg | 112219 | 132 |
| 998820 | 2016-08-05 | 70.1 | images (21).jpg | 112220 | 133 |
| 998821 | 2017-09-05 | 75.1 | images (22).jpg | 112221 | 133 |
| 998822 | 2018-10-05 | 78.1 | images (23).jpg | 112222 | 134 |
| 998823 | 2019-11-05 | 67.3 | images (24).jpg | 112223 | 134 |
| 998824 | 2020-12-05 | 69.3 | images (25).jpg | 112224 | 135 |
| 998825 | 2016-08-06 | 92.3 | images (26).jpg | 112225 | 135 |
| 998826 | 2017-09-06 | 68.3 | images (27).jpg | 112226 | 136 |
| 998827 | 2018-10-06 | 79.3 | images (28).jpg | 112227 | 136 |
| 998828 | 2019-11-06 | 86.9 | images (29).jpg | 112228 | 137 |
| 998829 | 2020-12-06 | 65 | images (30).jpg | 112229 | 137 |
| 998830 | 2016-08-07 | 66 | images (31).jpg | 112230 | 138 |
| 998831 | 2017-09-07 | 67 | images (32).jpg | 112231 | 138 |
| 998832 | 2018-10-07 | 68 | images (33).jpg | 112232 | 139 |
| 998833 | 2019-11-07 | 69 | images (34).jpg | 112233 | 139 |
| 998834 | 2020-12-07 | 70.2 | images (35).jpg | 112234 | 140 |
| 998835 | 2016-08-08 | 71.2 | images (36).jpg | 112235 | 140 |
| 998836 | 2017-09-08 | 72.2 | images (37).jpg | 112236 | 141 |
| 998837 | 2018-10-08 | 73.2 | images (38).jpg | 112237 | 141 |
| 998838 | 2019-11-08 | 74.2 | images (39).jpg | 112238 | 142 |
| 998839 | 2020-12-08 | 75.5 | images (40).jpg | 112239 | 142 |
| 998840 | 2016-08-09 | 76.7 | images (41).jpg | 112240 | 143 |
| 998841 | 2017-09-09 | 77.2 | images (42).jpg | 112241 | 143 |
| 998842 | 2018-10-09 | 78.5 | images (43).jpg | 112242 | 144 |
| 998843 | 2019-11-09 | 79.5 | images (44).jpg | 112243 | 144 |
| 998844 | 2020-12-09 | 99.4 | images (45).jpg | 112244 | 145 |
| 998845 | 2016-08-10 | 91.4 | images (46).jpg | 112245 | 145 |
| 998846 | 2017-09-10 | 92.4 | images (47).jpg | 112246 | 146 |
| 998847 | 2018-10-10 | 93.4 | images (48).jpg | 112247 | 146 |
| 998848 | 2019-11-10 | 94.4 | images (49).jpg | 112248 | 147 |
| 998849 | 2020-12-10 | 95.6 | images (50).jpg | 112249 | 147 |
| 998850 | 2016-08-11 | 96.6 | images (51).jpg | 112250 | 148 |
| 998851 | 2017-09-11 | 97.6 | images (52).jpg | 112251 | 148 |
| 998852 | 2018-10-11 | 88.6 | images (53).jpg | 112252 | 149 |
| 998853 | 2019-11-11 | 89.6 | images (54).jpg | 112253 | 149 |
| 998854 | 2020-12-11 | 90.6 | images (55).jpg | 112254 | 150 |
| 998855 | 2016-08-12 | 91.7 | images (56).jpg | 112255 | 150 |
| 998856 | 2017-09-12 | 92.2 | images (57).jpg | 112256 | 151 |
| 998857 | 2018-10-12 | 93.5 | images (58).jpg | 112257 | 151 |
| 998858 | 2019-11-12 | 94.5 | images (59).jpg | 112258 | 152 |
| 998859 | 2020-12-12 | 95.5 | images (60).jpg | 112259 | 152 |
| 998860 | 2016-08-13 | 96.7 | images (61).jpg | 112260 | 153 |
| 998861 | 2017-09-13 | 97.2 | images (62).jpg | 112261 | 153 |
| 998862 | 2018-10-13 | 98.5 | images (63).jpg | 112262 | 154 |
| 998863 | 2019-11-13 | 99.5 | images (64).jpg | 112263 | 154 |
| 998864 | 2020-12-13 | 99.1 | images (65).jpg | 112264 | 155 |
| 998865 | 2016-08-14 | 85.9 | images (66).jpg | 112265 | 155 |
| 998866 | 2017-09-14 | 80.9 | images (67).jpg | 112266 | 156 |
| 998867 | 2018-10-14 | 90.9 | images (68).jpg | 112267 | 156 |
| 998868 | 2019-11-14 | 91.9 | images (69).jpg | 112268 | 157 |
| 998869 | 2020-12-14 | 80.9 | images (70).jpg | 112269 | 157 |
| 998870 | 2016-08-15 | 73.9 | images (71).jpg | 112270 | 158 |
| 998871 | 2017-09-15 | 82.9 | images (72).jpg | 112271 | 158 |
| 998872 | 2018-10-15 | 95.9 | images (73).jpg | 112272 | 159 |
| 998873 | 2019-11-15 | 66.9 | images (74).jpg | 112273 | 159 |
| 998874 | 2020-12-15 | 89.9 | images (75).jpg | 112274 | 160 |
| 998875 | 2016-08-16 | 78.9 | images (76).jpg | 112275 | 160 |
| 998876 | 2017-09-16 | 99.9 | images (77).jpg | 112276 | 161 |
| 998877 | 2018-10-16 | 64.9 | images (78).jpg | 112277 | 161 |
| 998878 | 2019-11-16 | 68.9 | images (79).jpg | 112278 | 162 |
| 998879 | 2020-12-16 | 86.9 | images (80).jpg | 112279 | 162 |
| 998880 | 2016-08-17 | 84.7 | images (81).jpg | 112280 | 163 |
| 998881 | 2017-09-17 | 80.2 | images (82).jpg | 112281 | 163 |
| 998882 | 2018-10-17 | 93.5 | images (83).jpg | 112282 | 164 |
| 998883 | 2019-11-17 | 95.5 | images (84).jpg | 112283 | 164 |
| 998884 | 2020-12-17 | 99.5 | images (85).jpg | 112284 | 165 |
| 998885 | 2016-08-18 | 8.7 | images (86).jpg | 112285 | 165 |
| 998886 | 2017-09-18 | 83.2 | images (87).jpg | 112286 | 165 |
| 998887 | 2018-10-18 | 99.5 | images (88).jpg | 112287 | 166 |
| 998888 | 2019-11-18 | 97.5 | images (89).jpg | 112288 | 166 |
| 998889 | 2020-12-18 | 93.5 | images (90).jpg | 112289 | 166 |
| 998890 | 2016-08-19 | 91.7 | images (91).jpg | 112290 | 167 |
| 998891 | 2017-09-19 | 92.2 | images (92).jpg | 112291 | 167 |
| 998892 | 2018-10-19 | 93.5 | images (93).jpg | 112292 | 167 |
| 998893 | 2019-11-19 | 94.5 | images (94).jpg | 112293 | 168 |
| 998894 | 2020-12-19 | 95.5 | images (95).jpg | 112294 | 168 |
| 998895 | 2016-08-20 | 96.7 | images (96).jpg | 112295 | 168 |
| 998896 | 2017-09-20 | 91.2 | images (97).jpg | 112296 | 169 |
| 998897 | 2018-10-20 | 93.5 | images (98).jpg | 112297 | 169 |
| 998898 | 2019-11-20 | 96.5 | images (99).jpg | 112298 | 169 |
| 998899 | 2020-12-20 | 91.5 | images (100).jpg | 112299 | 169 |

**SQL Code:**

use userDB2

drop table Enrollment;

drop table Student;

drop table Department;

drop table College;

drop table University;

drop table users;

create table users(

userName nvarchar(50),

[password] nvarchar(50)

)

go

insert into users values ('admin','1234');

insert into users values ('osama','1234');

insert into users values ('ali','4321');

insert into users values ('hozan','999999999');

insert into users values ('hozan','999');

create table University(

Univ\_ID int primary key,

Univ\_Name nvarchar(50),

Adress nvarchar(50)

)

go

insert into University values(1,'Salahaddin','Erbil-karkuk-street');

insert into University values(2,'Sulaimani','Sulaimani-street');

insert into University values(3,'Duhok','Duho-Zakho-street');

select \* from University;

create table College(

College\_ID int primary key,

College\_Name nvarchar(50),

Dean nvarchar(50),

Univ\_ID int foreign key references University(Univ\_ID)

)

go

insert into College values(11,'Science','Professor Hauresh',1);

insert into College values(12,'Engineering','Professor Hamdin',1);

insert into College values(13,'Language','Professor Ahmad',1);

insert into College values(14,'Engineering','Professor Ramadan',2);

insert into College values(15,'Medicine','Professor Ali',2);

insert into College values(16,'Language','Professor Sarkheel',3);

insert into College values(17,'Science','Professor Shekhmus',3);

insert into College values(18,'Engineering','Professor Zana',3);

insert into College values(19,'Medicine','Professor Aland',3);

select \* from College;

create table Department(

Dep\_ID int primary key,

Dep\_Name nvarchar(50),

Supervisor nvarchar(50),

College\_ID int foreign key references College(College\_ID)

)

go

insert into Department values(123,'CS & IT','Dr Dler',11);

insert into Department values(124,'Math','Dr Maria',11);

insert into Department values(125,'Physic','Dr Nushi',11);

insert into Department values(126,'Geology','Dr Mohammed',11);

insert into Department values(127,'Biology','Dr Jose',11);

insert into Department values(128,'chemic','Dr Wei',11);

insert into Department values(129,'Software Engineering','Dr Salar',12);

insert into Department values(130,'Mechanical & Mechatronics','Dr Yan',12);

insert into Department values(131,'Architecture','Dr John',12);

insert into Department values(132,'Civil','Dr David',12);

insert into Department values(133,'Electrical','Dr Li',12);

insert into Department values(134,'Kurdish','Dr Eanoo',13);

insert into Department values(135,'Arabic','Dr Abdul',13);

insert into Department values(136,'English','Dr Ana',13);

insert into Department values(137,'French','Dr Ying',13);

insert into Department values(138,'Germany','Dr Michael',13);

insert into Department values(139,'Turkish','Dr Juan',13);

insert into Department values(140,'Software Engineering','Dr Anna',14);

insert into Department values(141,'Mechanical & Mechatronics','Dr Mary',14);

insert into Department values(142,'Architecture','Dr Jean',14);

insert into Department values(143,'Civil','Dr Robert',14);

insert into Department values(144,'Electrical','Dr Daniel',14);

insert into Department values(145,'General Medicine','Dr Luis',15);

insert into Department values(146,'Dentistry','Dr Carlos',15);

insert into Department values(147,'Pharmacy','Dr James',15);

insert into Department values(148,'Ophthalmology','Dr Antonio',15);

insert into Department values(149,'Kurdish','Dr Joseph',16);

insert into Department values(150,'Arabic','Dr Hui',16);

insert into Department values(151,'English','Dr Elena',16);

insert into Department values(152,'French','Dr Francisco',16);

insert into Department values(153,'Germany','Dr Hong',16);

insert into Department values(154,'Turkish','Dr Marie',16);

insert into Department values(155,'CS & IT','Dr Min',17);

insert into Department values(156,'Math','Dr Lei',17);

insert into Department values(157,'Physic','Dr Yu',17);

insert into Department values(158,'Geology','Dr Ibrahim',17);

insert into Department values(159,'Biology','Dr Peter',17);

insert into Department values(160,'chemic','Dr Fatima',17);

insert into Department values(161,'Software Engineering','Dr Aleksandr',18);

insert into Department values(162,'Mechanical & Mechatronics','Dr Richard',18);

insert into Department values(163,'Architecture','Dr Xin',18);

insert into Department values(164,'Civil','Dr Bin',18);

insert into Department values(165,'Electrical','Dr Paul',18);

insert into Department values(166,'General Medicine','Dr Ping',19);

insert into Department values(167,'Dentistry','Dr Lin',19);

insert into Department values(168,'Pharmacy','Dr Olga',19);

insert into Department values(169,'Ophthalmology','Dr Sri',19);

select \* from Department;

create table Student(

Student\_ID bigint primary key,

First\_Name nvarchar(50),

Last\_Name nvarchar(50),

Gender nvarchar(10),

Date\_Of\_Birth date,

Email nvarchar(50),

Phone\_Number nvarchar(50)

)

go

insert into Student values(112200,'James','John','Male','1999-7-1','James@gmail.com','+9647826030525');

insert into Student values(112201,'Mary','Patricia','Female','2000-7-2','Mary@gmail.com','+9647835459526');

insert into Student values(112202,'Robert','Michael','Male','2000-7-3','Robert@gmail.com','+9647507788527');

insert into Student values(112203,'Jennifer','Linda','Female','2000-7-4','Jennifer@gmail.com','+9647835459528');

insert into Student values(112204,'William','David','Male','1999-7-5','William@gmail.com','+9647826030529');

insert into Student values(112205,'Elizabeth','Barbara','Female','2000-7-6','Elizabeth@gmail.com','+9647835459530');

insert into Student values(112206,'Richard','Joseph','Male','2000-7-7','Richard@gmail.com','+9647507788531');

insert into Student values(112207,'Susan','Jessica','Female','2000-7-8','Susan@gmail.com','+9647835459532');

insert into Student values(112208,'Thomas','Charles','Male','1999-7-9','Thomas@gmail.com','+9647826030533');

insert into Student values(112209,'Sarah','Karen','Female','2000-7-10','Sarah@gmail.com','+9647835459534');

insert into Student values(112210,'Christopher','Daniel','Male','2000-7-11','Christopher@gmail.com','+9647507788535');

insert into Student values(112211,'Nancy','Lisa','Female','2000-7-12','Nancy@gmail.com','+9647835459536');

insert into Student values(112212,'Matthew','Anthony','Male','1999-7-13','Matthew@gmail.com','+9647826030537');

insert into Student values(112213,'Margaret','Betty','Female','2000-7-14','Margaret@gmail.com','+9647835459538');

insert into Student values(112214,'Donald','Mark','Male','2000-7-15','Donald@gmail.com','+9647507788539');

insert into Student values(112215,'Sandra','Ashley','Female','2000-7-16','Sandra@gmail.com','+9647835459540');

insert into Student values(112216,'Paul','Steven','Male','1999-7-17','Paul@gmail.com','+9647826030541');

insert into Student values(112217,'Dorothy','Kimberly','Female','2000-7-18','Dorothy@gmail.com','+9647835459542');

insert into Student values(112218,'Andrew','Kenneth','Male','2000-7-19','Andrew@gmail.com','+9647507788543');

insert into Student values(112219,'Emily','Donna','Female','2000-7-20','Emily@gmail.com','+9647835459544');

insert into Student values(112220,'Joshua','Kevin','Male','1999-7-21','Joshua@gmail.com','+9647826030545');

insert into Student values(112221,'Michelle','Carol','Female','2000-7-22','Michelle@gmail.com','+9647835459546');

insert into Student values(112222,'Brian','George','Male','2000-7-23','Brian@gmail.com','+9647507788547');

insert into Student values(112223,'Amanda','Melissa','Female','2000-7-24','Amanda@gmail.com','+9647835459548');

insert into Student values(112224,'Edward','Ronald','Male','1999-7-25','Edward@gmail.com','+9647826030549');

insert into Student values(112225,'Deborah','Stephanie','Female','2000-8-1','Deborah@gmail.com','+9647835459550');

insert into Student values(112226,'Timothy','Jason','Male','2000-8-2','Timothy@gmail.com','+9647507788551');

insert into Student values(112227,'Rebecca','Laura','Female','2000-8-3','Rebecca@gmail.com','+9647835459552');

insert into Student values(112228,'Jeffrey','Ryan','Male','1999-8-4','Jeffrey@gmail.com','+9647826030553');

insert into Student values(112229,'Sharon','Cynthia','Female','2000-8-5','Sharon@gmail.com','+9647835459554');

insert into Student values(112230,'Jacob','Gary','Male','2000-8-6','Jacob@gmail.com','+9647507788555');

insert into Student values(112231,'Kathleen','Amy','Female','2000-8-7','Kathleen@gmail.com','+9647835459556');

insert into Student values(112232,'Nicholas','Eric','Male','1999-8-8','Nicholas@gmail.com','+9647826030557');

insert into Student values(112233,'Shirley','Angela','Female','2000-8-9','Shirley@gmail.com','+9647835459558');

insert into Student values(112234,'Jonathan','Stephen','Male','2000-8-10','Jonathan@gmail.com','+9647507788559');

insert into Student values(112235,'Helen','Anna','Female','2000-8-11','Helen@gmail.com','+9647835459560');

insert into Student values(112236,'Larry','Justin','Male','1999-8-12','Larry@gmail.com','+9647826030561');

insert into Student values(112237,'Brenda','Pamela','Female','2000-8-13','Brenda@gmail.com','+9647835459562');

insert into Student values(112238,'Scott','Brandon','Male','2000-8-14','Scott@gmail.com','+9647507788563');

insert into Student values(112239,'Nicole','Samantha','Female','2000-8-15','Nicole@gmail.com','+9647835459564');

insert into Student values(112240,'Benjamin','Samuel','Male','1999-8-16','Benjamin@gmail.com','+9647826030565');

insert into Student values(112241,'Katherine','Emma','Female','2000-8-17','Katherine@gmail.com','+9647835459566');

insert into Student values(112242,'Frank','Gregory','Male','2000-8-18','Frank@gmail.com','+9647507788567');

insert into Student values(112243,'Ruth','Christine','Female','2000-8-19','Ruth@gmail.com','+9647835459568');

insert into Student values(112244,'Raymond  ','Alexander','Male','1999-8-20','Raymond@gmail.com','+9647826030569');

insert into Student values(112245,'Catherine','Debra','Female','2000-8-21','Catherine@gmail.com','+9647835459570');

insert into Student values(112246,'Patrick','Jack','Male','2000-8-22','Patrick@gmail.com','+9647507788571');

insert into Student values(112247,'Rachel','Carolyn','Female','2000-8-23','Rachel@gmail.com','+9647835459572');

insert into Student values(112248,'Dennis','Jerry','Male','1999-8-24','Dennis@gmail.com','+9647826030573');

insert into Student values(112249,'Janet','Virginia','Female','2000-8-25','Janet@gmail.com','+9647835459574');

insert into Student values(112250,'Tyler','Aaron','Male','2000-9-1','Tyler@gmail.com','+9647507788575');

insert into Student values(112251,'Maria','Heather','Female','2000-9-2','Maria@gmail.com','+9647835459576');

insert into Student values(112252,'Jose','Henry','Male','1999-9-3','Jose@gmail.com','+9647826030577');

insert into Student values(112253,'Diane','Julie','Female','2000-9-4','Diane@gmail.com','+9647835459578');

insert into Student values(112254,'Adam','Douglas','Male','2000-9-5','Adam@gmail.com','+9647507788579');

insert into Student values(112255,'Joyce','Victoria','Female','2000-9-6','Joyce@gmail.com','+9647835459580');

insert into Student values(112256,'Nathan','Peter','Male','1999-9-7','Nathan@gmail.com','+9647826030581');

insert into Student values(112257,'Kelly','Christina','Female','2000-9-8','Kelly@gmail.com','+9647835459582');

insert into Student values(112258,'Zachary','Kyle','Male','2000-9-9','Zachary@gmail.com','+9647507788583');

insert into Student values(112259,'Lauren','Joan','Female','2000-9-10','Lauren@gmail.com','+9647835459584');

insert into Student values(112260,'Walter','Harold','Male','1999-9-11','Walter@gmail.com','+9647826030585');

insert into Student values(112261,'Evelyn','Olivia','Female','2000-9-12','Evelyn@gmail.com','+9647835459586');

insert into Student values(112262,'Jeremy','Ethan','Male','2000-9-13','Jeremy@gmail.com','+9647507788587');

insert into Student values(112263,'Judith','Megan','Female','2000-9-14','Judith@gmail.com','+9647835459588');

insert into Student values(112264,'Carl','Keith','Male','1999-9-15','Carl@gmail.com','+9647826030589');

insert into Student values(112265,'Cheryl','Martha','Female','2000-9-16','Cheryl@gmail.com','+9647835459590');

insert into Student values(112266,'Roger','Gerald','Male','2000-9-17','Roger@gmail.com','+9647507788591');

insert into Student values(112267,'Andrea','Frances','Female','2000-9-18','Andrea@gmail.com','+9647835459592');

insert into Student values(112268,'Christian','Terry','Male','1999-9-19','Christian@gmail.com','+9647826030593');

insert into Student values(112269,'Hannah','Jacqueline','Female','2000-9-20','Hannah@gmail.com','+9647835459594');

insert into Student values(112270,'Sean','Arthur','Male','2000-9-21','Sean@gmail.com','+9647507788595');

insert into Student values(112271,'Ann','Gloria','Female','2000-9-22','Ann@gmail.com','+9647835459596');

insert into Student values(112272,'Austin','Noah','Male','1999-9-23','Austin@gmail.com','+9647826030597');

insert into Student values(112273,'Jean','Kathryn','Female','2000-9-24','Jean@gmail.com','+9647835459598');

insert into Student values(112274,'Lawrence','Jesse','Male','2000-9-25','Lawrence@gmail.com','+9647507788599');

insert into Student values(112275,'Alice','Teresa','Female','2000-10-1','Alice@gmail.com','+9647835459600');

insert into Student values(112276,'Joe','Bryan','Male','1999-10-2','Joe@gmail.com','+9647826030601');

insert into Student values(112277,'Sara','Janice','Female','2000-10-3','Sara@gmail.com','+9647835459602');

insert into Student values(112278,'Billy','Jordan','Male','2000-10-4','Billy@gmail.com','+9647507788603');

insert into Student values(112279,'Doris','Madison','Female','2000-10-5','Doris@gmail.com','+9647835459604');

insert into Student values(112280,'Albert','Dylan','Male','1999-10-6','Albert@gmail.com','+9647826030605');

insert into Student values(112281,'Julia','Grace','Female','2000-10-7','Julia@gmail.com','+9647835459607');

insert into Student values(112282,'Bruce','Willie','Male','2000-10-8','Bruce@gmail.com','+9647507788608');

insert into Student values(112283,'Judy','Abigail','Female','2000-10-9','Judy@gmail.com','+9647835459609');

insert into Student values(112284,'Gabriel','Alan','Male','1999-10-10','Gabriel@gmail.com','+9647826030610');

insert into Student values(112285,'Marie','Denise','Female','2000-10-11','Marie@gmail.com','+9647835459611');

insert into Student values(112286,'Juan','Logan','Male','2000-10-12','Juan@gmail.com','+9647507788612');

insert into Student values(112287,'Beverly','Amber','Female','2000-10-13','Beverly@gmail.com','+9647835459613');

insert into Student values(112288,'Wayne','Ralph','Male','1999-10-14','Wayne@gmail.com','+9647826030614');

insert into Student values(112289,'Theresa','Marilyn','Female','2000-10-15','Theresa@gmail.com','+9647835459615');

insert into Student values(112290,'Roy','Eugene','Male','2000-10-16','Roy@gmail.com','+9647507788616');

insert into Student values(112291,'Danielle','Diana','Female','2000-10-17','Danielle@gmail.com','+9647835459617');

insert into Student values(112292,'Randy','Vincent','Male','1999-10-18','Randy@gmail.com','+9647826030618');

insert into Student values(112293,'Brittany','Natalie','Female','2000-10-19','Brittany@gmail.com','+9647835459619');

insert into Student values(112294,'Russell','Louis','Male','2000-10-20','Russell@gmail.com','+9647507788620');

insert into Student values(112295,'Sophia','Rose','Female','2000-10-21','Sophia@gmail.com','+9647835459621');

insert into Student values(112296,'Philip','Bobby','Male','1999-10-22','Philip@gmail.com','+9647826030622');

insert into Student values(112297,'Isabella','Alexis','Female','2000-10-23','Isabella@gmail.com','+9647835459623');

insert into Student values(112298,'Kayla','Charlotte','Male','2000-10-24','Kayla@gmail.com','+9647507788624');

insert into Student values(112299,'shereen','sulaiman','Female','2000-10-25','shirin2021@gmail.com','+9647835459625');

select \* from Student;

create table Enrollment(

Enrol\_ID bigint primary key,

Enrollment\_Date date,

Grade float,

Image\_Certificate nvarchar(max),

Student\_ID bigint unique foreign key references Student(Student\_ID),

Dep\_ID int foreign key references Department(Dep\_ID)

)

go

insert into Enrollment values(998800,'2016-8-1',65.2,'images (1).jpg',112200,123);

insert into Enrollment values(998801,'2017-9-1',70.9,'images (2).jpg',112201,123);

insert into Enrollment values(998802,'2018-10-1',80.7,'images (3).jpg',112202,124);

insert into Enrollment values(998803,'2019-11-1',90.4,'images (4).jpg',112203,124);

insert into Enrollment values(998804,'2020-12-1',92.6,'images (5).jpg',112204,125);

insert into Enrollment values(998805,'2016-8-2',86.7,'images (6).jpg',112205,125);

insert into Enrollment values(998806,'2017-9-2',80.2,'images (7).jpg',112206,126);

insert into Enrollment values(998807,'2018-10-2',60.5,'images (8).jpg',112207,126);

insert into Enrollment values(998808,'2019-11-2',50.5,'images (9).jpg',112208,127);

insert into Enrollment values(998809,'2020-12-2',55.5,'images (10).jpg',112209,127);

insert into Enrollment values(998810,'2016-8-3',66.7,'images (11).jpg',112210,128);

insert into Enrollment values(998811,'2017-9-3',77.2,'images (12).jpg',112211,128);

insert into Enrollment values(998812,'2018-10-3',90.5,'images (13).jpg',112212,129);

insert into Enrollment values(998813,'2019-11-3',88.5,'images (14).jpg',112213,129);

insert into Enrollment values(998814,'2020-12-3',99.0,'images (15).jpg',112214,130);

insert into Enrollment values(998815,'2016-8-4',81.4,'images (16).jpg',112215,130);

insert into Enrollment values(998816,'2017-9-4',74.3,'images (17).jpg',112216,131);

insert into Enrollment values(998817,'2018-10-4',91.8,'images (18).jpg',112217,131);

insert into Enrollment values(998818,'2019-11-4',92.8,'images (19).jpg',112218,132);

insert into Enrollment values(998819,'2020-12-4',93.8,'images (20).jpg',112219,132);

insert into Enrollment values(998820,'2016-8-5',70.1,'images (21).jpg',112220,133);

insert into Enrollment values(998821,'2017-9-5',75.1,'images (22).jpg',112221,133);

insert into Enrollment values(998822,'2018-10-5',78.1,'images (23).jpg',112222,134);

insert into Enrollment values(998823,'2019-11-5',67.3,'images (24).jpg',112223,134);

insert into Enrollment values(998824,'2020-12-5',69.3,'images (25).jpg',112224,135);

insert into Enrollment values(998825,'2016-8-6',92.3,'images (26).jpg',112225,135);

insert into Enrollment values(998826,'2017-9-6',68.3,'images (27).jpg',112226,136);

insert into Enrollment values(998827,'2018-10-6',79.3,'images (28).jpg',112227,136);

insert into Enrollment values(998828,'2019-11-6',86.9,'images (29).jpg',112228,137);

insert into Enrollment values(998829,'2020-12-6',65.0,'images (30).jpg',112229,137);

insert into Enrollment values(998830,'2016-8-7',66.0,'images (31).jpg',112230,138);

insert into Enrollment values(998831,'2017-9-7',67.0,'images (32).jpg',112231,138);

insert into Enrollment values(998832,'2018-10-7',68.0,'images (33).jpg',112232,139);

insert into Enrollment values(998833,'2019-11-7',69.0,'images (34).jpg',112233,139);

insert into Enrollment values(998834,'2020-12-7',70.2,'images (35).jpg',112234,140);

insert into Enrollment values(998835,'2016-8-8',71.2,'images (36).jpg',112235,140);

insert into Enrollment values(998836,'2017-9-8',72.2,'images (37).jpg',112236,141);

insert into Enrollment values(998837,'2018-10-8',73.2,'images (38).jpg',112237,141);

insert into Enrollment values(998838,'2019-11-8',74.2,'images (39).jpg',112238,142);

insert into Enrollment values(998839,'2020-12-8',75.5,'images (40).jpg',112239,142);

insert into Enrollment values(998840,'2016-8-9',76.7,'images (41).jpg',112240,143);

insert into Enrollment values(998841,'2017-9-9',77.2,'images (42).jpg',112241,143);

insert into Enrollment values(998842,'2018-10-9',78.5,'images (43).jpg',112242,144);

insert into Enrollment values(998843,'2019-11-9',79.5,'images (44).jpg',112243,144);

insert into Enrollment values(998844,'2020-12-9',99.4,'images (45).jpg',112244,145);

insert into Enrollment values(998845,'2016-8-10',91.4,'images (46).jpg',112245,145);

insert into Enrollment values(998846,'2017-9-10',92.4,'images (47).jpg',112246,146);

insert into Enrollment values(998847,'2018-10-10',93.4,'images (48).jpg',112247,146);

insert into Enrollment values(998848,'2019-11-10',94.4,'images (49).jpg',112248,147);

insert into Enrollment values(998849,'2020-12-10',95.6,'images (50).jpg',112249,147);

insert into Enrollment values(998850,'2016-8-11',96.6,'images (51).jpg',112250,148);

insert into Enrollment values(998851,'2017-9-11',97.6,'images (52).jpg',112251,148);

insert into Enrollment values(998852,'2018-10-11',88.6,'images (53).jpg',112252,149);

insert into Enrollment values(998853,'2019-11-11',89.6,'images (54).jpg',112253,149);

insert into Enrollment values(998854,'2020-12-11',90.6,'images (55).jpg',112254,150);

insert into Enrollment values(998855,'2016-8-12',91.7,'images (56).jpg',112255,150);

insert into Enrollment values(998856,'2017-9-12',92.2,'images (57).jpg',112256,151);

insert into Enrollment values(998857,'2018-10-12',93.5,'images (58).jpg',112257,151);

insert into Enrollment values(998858,'2019-11-12',94.5,'images (59).jpg',112258,152);

insert into Enrollment values(998859,'2020-12-12',95.5,'images (60).jpg',112259,152);

insert into Enrollment values(998860,'2016-8-13',96.7,'images (61).jpg',112260,153);

insert into Enrollment values(998861,'2017-9-13',97.2,'images (62).jpg',112261,153);

insert into Enrollment values(998862,'2018-10-13',98.5,'images (63).jpg',112262,154);

insert into Enrollment values(998863,'2019-11-13',99.5,'images (64).jpg',112263,154);

insert into Enrollment values(998864,'2020-12-13',99.1,'images (65).jpg',112264,155);

insert into Enrollment values(998865,'2016-8-14',85.9,'images (66).jpg',112265,155);

insert into Enrollment values(998866,'2017-9-14',80.9,'images (67).jpg',112266,156);

insert into Enrollment values(998867,'2018-10-14',90.9,'images (68).jpg',112267,156);

insert into Enrollment values(998868,'2019-11-14',91.9,'images (69).jpg',112268,157);

insert into Enrollment values(998869,'2020-12-14',80.9,'images (70).jpg',112269,157);

insert into Enrollment values(998870,'2016-8-15',73.9,'images (71).jpg',112270,158);

insert into Enrollment values(998871,'2017-9-15',82.9,'images (72).jpg',112271,158);

insert into Enrollment values(998872,'2018-10-15',95.9,'images (73).jpg',112272,159);

insert into Enrollment values(998873,'2019-11-15',66.9,'images (74).jpg',112273,159);

insert into Enrollment values(998874,'2020-12-15',89.9,'images (75).jpg',112274,160);

insert into Enrollment values(998875,'2016-8-16',78.9,'images (76).jpg',112275,160);

insert into Enrollment values(998876,'2017-9-16',99.9,'images (77).jpg',112276,161);

insert into Enrollment values(998877,'2018-10-16',64.9,'images (78).jpg',112277,161);

insert into Enrollment values(998878,'2019-11-16',68.9,'images (79).jpg',112278,162);

insert into Enrollment values(998879,'2020-12-16',86.9,'images (80).jpg',112279,162);

insert into Enrollment values(998880,'2016-8-17',84.7,'images (81).jpg',112280,163);

insert into Enrollment values(998881,'2017-9-17',80.2,'images (82).jpg',112281,163);

insert into Enrollment values(998882,'2018-10-17',93.5,'images (83).jpg',112282,164);

insert into Enrollment values(998883,'2019-11-17',95.5,'images (84).jpg',112283,164);

insert into Enrollment values(998884,'2020-12-17',99.5,'images (85).jpg',112284,165);

insert into Enrollment values(998885,'2016-8-18',8.7,'images (86).jpg',112285,165);

insert into Enrollment values(998886,'2017-9-18',83.2,'images (87).jpg',112286,165);

insert into Enrollment values(998887,'2018-10-18',99.5,'images (88).jpg',112287,166);

insert into Enrollment values(998888,'2019-11-18',97.5,'images (89).jpg',112288,166);

insert into Enrollment values(998889,'2020-12-18',93.5,'images (90).jpg',112289,166);

insert into Enrollment values(998890,'2016-8-19',91.7,'images (91).jpg',112290,167);

insert into Enrollment values(998891,'2017-9-19',92.2,'images (92).jpg',112291,167);

insert into Enrollment values(998892,'2018-10-19',93.5,'images (93).jpg',112292,167);

insert into Enrollment values(998893,'2019-11-19',94.5,'images (94).jpg',112293,168);

insert into Enrollment values(998894,'2020-12-19',95.5,'images (95).jpg',112294,168);

insert into Enrollment values(998895,'2016-8-20',96.7,'images (96).jpg',112295,168);

insert into Enrollment values(998896,'2017-9-20',91.2,'images (97).jpg',112296,169);

insert into Enrollment values(998897,'2018-10-20',93.5,'images (98).jpg',112297,169);

insert into Enrollment values(998898,'2019-11-20',96.5,'images (99).jpg',112298,169);

insert into Enrollment values(998899,'2020-12-20',91.5,'images (100).jpg',112299,169);

select \* from Enrollment;

SELECT \* FROM University u

join College c

on u.Univ\_ID=c.Univ\_ID

join Department d

on c.College\_ID=d.College\_ID

join Enrollment e

on d.Dep\_ID=e.Dep\_ID

join Student s

on e.Student\_ID=s.Student\_ID;

-----------------------------------------------------------

**Python Code:**

# Our Project ==> Student Enrollment System

# it will return the location

from tkinter import filedialog

from PIL import ImageTk, Image

from tkinter import \*

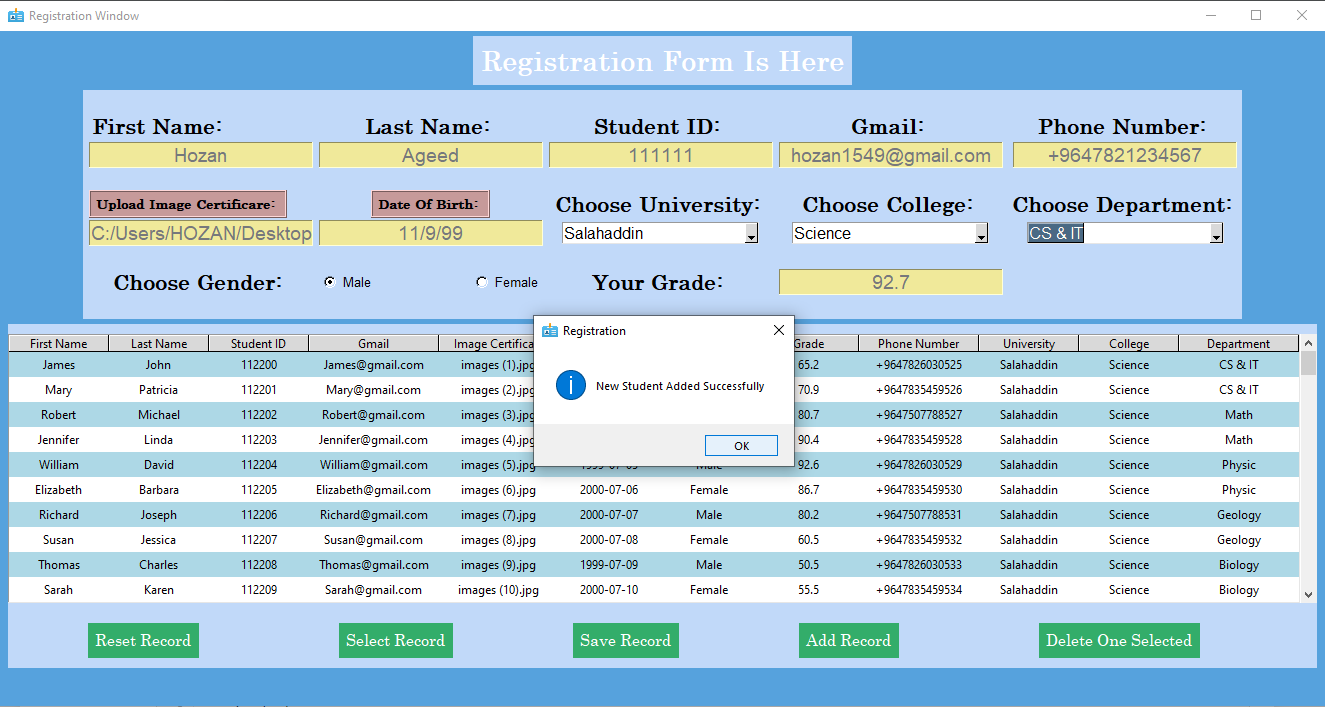
from datetime import date

from tkcalendar import \*

from tkinter import ttk

import pyodbc

from tkinter import messagebox



def dashBoard():  # Registration Form

    root2.destroy()

    root.destroy()

    global regis\_form\_window

    regis\_form\_window = Tk()  # New Window

    regis\_form\_window.geometry("1325x675+20+0")

    regis\_form\_window.title("Registration Window")

    regis\_form\_window.config(bg="#56A2DD")

#     regis\_form\_window.resizable(False,False)

    regis\_form\_window.iconbitmap(r'student icon.ico')

    regis\_form\_window.focus\_force()

    l\_title = Label(regis\_form\_window,

                    text="Registration Form Is Here",

                    font=('Century', 20, 'bold'),

                    bg="#C1D9F9", fg="white")

    l\_title.pack(pady=5, ipadx=5, ipady=5)

    regis\_window = Frame(regis\_form\_window, bg="#C1D9F9")

    regis\_window.pack(anchor=N)

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=0)

    l\_first\_name = Label(regis\_window, text="First Name: ",

    font=('Century', 16, 'bold'),

    bg="#C1D9F9").grid(row=1, column=1, sticky=W)

    e\_first\_name = Entry(regis\_window, font=(

        'Arial', 14), fg="#76787C", bg="#f0e99a",

         relief="groove", justify="center")

    e\_first\_name.grid(row=2, column=1)

    e\_first\_name.focus\_force()

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=2)

    l\_last\_name = Label(regis\_window, text="Last Name: ",

     font=('Century', 16, 'bold'),

                        bg="#C1D9F9").grid(row=1, column=3)

    e\_last\_name = Entry(regis\_window, font=(

        'Arial', 14), fg="#76787C", bg="#f0e99a",

         relief="groove",justify="center")

    e\_last\_name.grid(row=2, column=3)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=4)

    l\_id = Label(regis\_window, text="Student ID: ",

     font=('Century', 16, 'bold'),

     bg="#C1D9F9").grid(row=1, column=5)

    e\_id = Entry(regis\_window, font=('Arial', 14),

     fg="#76787C",bg="#f0e99a",

      relief="groove",justify="center")

    e\_id.grid(row=2, column=5)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=6)

    l\_gmail = Label(regis\_window, text="Gmail: ", font=('Century', 16, 'bold'),

                    bg="#C1D9F9").grid(row=1, column=7)

    e\_gmail = Entry(regis\_window, font=('Arial', 14), fg="#76787C",

                    bg="#f0e99a", relief="groove", justify="center")

    e\_gmail.grid(row=2, column=7)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=8)

    l\_phone = Label(regis\_window, text="Phone Number: ",

     font=('Century', 16, 'bold'),bg="#C1D9F9").grid(row=1, column=9)

    e\_phone = Entry(regis\_window, font=('Arial', 14), fg="#76787C",

                    bg="#f0e99a", relief="groove", justify="center")

    e\_phone.grid(row=2, column=9)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=3, column=0)  # space for row

    def open\_file():

        regis\_window.filename = filedialog.askopenfilename(initialdir="C:\\Users\\HOZAN\\Desktop",

         title="Select A File",filetypes=(("png files", "\*.png"),

         ("all files", "\*.\*")))

        v\_image.set(regis\_window.filename)

    b\_upload = Button(regis\_window, text="Upload Image Certificare: ",

     font=('Century', 10, 'bold'), cursor="hand2",

                      bg="#C69A9A", relief="ridge",

                       command=open\_file).grid(row=4, column=1, sticky=W)

    v\_image = StringVar()

    v\_image.set("")

    e\_upload = Entry(regis\_window, textvariable=v\_image, font=(

        'Arial', 14), fg="#76787C", bg="#f0e99a", relief="groove")

    e\_upload.grid(row=5, column=1)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=2)

    def get\_date():       # global v\_date

        v\_date.set(cal.get\_date())

        date\_window.destroy()

    def grab\_date():

        global date\_window

        date\_window = Toplevel(regis\_window)

        date\_window.title("Date Window")

        date\_window.focus\_force()

        global cal

        cal = Calendar(date\_window, selectmode="day",

                       year=1999, month=11, day=9)

        cal.pack(pady=20)

        date\_button = Button(date\_window, text="Get Date", command=get\_date)

        date\_button.pack(pady=5)

        date\_window.mainloop()

    v\_date = StringVar()

    v\_date.set("")

    b\_Date = Button(regis\_window, text="Date Of Birth: ",

     font=('Century', 10, 'bold'), cursor="hand2",

                    bg="#C69A9A", relief="ridge",

                     command=grab\_date).grid(row=4, column=3)

    e\_Date = Entry(regis\_window, textvariable=v\_date,

     font=('Arial', 14),fg="#76787C", bg="#f0e99a",

      relief="groove", justify="center")

    e\_Date.grid(row=5, column=3)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=4)

    # Creating a list of Universities

    universities = [

        "Salahaddin",

        "Sulaimani",

        "Duhok"

    ]

    coll\_of\_sala = [

        "Science",

        "Engineering",

        "Language"

    ]

    coll\_of\_sulai = [

        "Medicine",

        "Engineering"

    ]

    # Creating a list of colleges

    colleges = [

        "Science",

        "Engineering",

        "Language",

        "Medicine"

    ]

    # Creating a list of departments of Science

    dep\_of\_science = [

        "CS & IT",

        "Math",

        "Physic",

        "Geology",

        "Biology",

        "chemic"

    ]

    # Creating a list of departments of Engineering

    dep\_of\_engineering = [

        "Software Engineering",

        "Mechanical & Mechatronics",

        "Architecture",

        "Civil",

        "Electrical"

    ]

    # Creating a list of departments of Language

    dep\_of\_language = [

        "Kurdish",

        "Arabic",

        "English",

        "French",

        "Germany",

        "Turkish"

    ]

    # Creating a list of departments of Medicine

    dep\_of\_Medicine = [

        "General Medicine",

        "Dentistry",

        "Pharmacy",

        "Ophthalmology"

    ]

    def pick\_dep(e):

        if coll\_combo.get() == "Science":

            dep\_combo.config(value=dep\_of\_science)

#             dep\_combo.current(0)

        elif coll\_combo.get() == "Engineering":

            dep\_combo.config(value=dep\_of\_engineering)

#             dep\_combo.current(0)

        elif coll\_combo.get() == "Language":

            dep\_combo.config(value=dep\_of\_language)

#             dep\_combo.current(0)

        elif coll\_combo.get() == "Medicine":

            dep\_combo.config(value=dep\_of\_Medicine)

#             dep\_combo.current(0)

    def pick\_coll(e):

        if univ\_combo.get() == "Salahaddin":

            coll\_combo.config(value=coll\_of\_sala)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

        elif univ\_combo.get() == "Sulaimani":

            coll\_combo.config(value=coll\_of\_sulai)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

        elif univ\_combo.get() == "Duhok":

            coll\_combo.config(value=colleges)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

    l\_univ = Label(regis\_window, text="Choose University: ",

     font=('Century', 16, 'bold'),

                   bg="#C1D9F9").grid(row=4, column=5)

    # create a dropdown box

    univ\_combo = ttk.Combobox(

        regis\_window, value=universities, font=('Arial', 12))

#     univ\_combo.current(0)

    univ\_combo.grid(row=5, column=5)

    univ\_combo.bind("<<ComboboxSelected>>", pick\_coll)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=6)

    l\_coll = Label(regis\_window, text="Choose College: ",

     font=('Century', 16, 'bold'),

     bg="#C1D9F9").grid(row=4, column=7)

    coll\_combo = ttk.Combobox(regis\_window,

     value=colleges, font=('Arial', 12))

#     coll\_combo.current(0)

    coll\_combo.grid(row=5, column=7)

    coll\_combo.bind("<<ComboboxSelected>>", pick\_dep)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=0, column=8)

    l\_dep = Label(regis\_window, text="Choose Department: ",

     font=('Century', 16, 'bold'),

                  bg="#C1D9F9").grid(row=4, column=9)

    # Department combo box

    dep\_combo = ttk.Combobox(regis\_window, value=[" "],

     font=('Arial', 12))

    dep\_combo.current(0)

    dep\_combo.grid(row=5, column=9)

    # -----------------------------------------------------------------------------------

    Label(regis\_window, bg="#C1D9F9").grid(row=6, column=0)  # space for row

    l\_gender = Label(regis\_window, text="Choose Gender: ",

     font=('Century', 16, 'bold'),

                     bg="#C1D9F9").grid(row=7, column=1)

    va = StringVar()

    va.set(" ")

    Radiobutton(regis\_window, text="Male", variable=va,

     value="Male",font=('Arial', 10),

      bg="#C1D9F9").grid(row=7, column=3, sticky=W)

    Radiobutton(regis\_window, text="Female", variable=va,

     value="Female",font=('Arial', 10),

      bg="#C1D9F9").grid(row=7, column=3, sticky=E)

    # -----------------------------------------------------------------------------------

    # space for row

    Label(regis\_window, bg="#C1D9F9").grid(row=8, column=0)

    l\_grade = Label(regis\_window, text="Your Grade: ",

     font=('Century', 16, 'bold'),

                    bg="#C1D9F9").grid(row=7, column=5)

    e\_grade = Entry(regis\_window, font=('Arial', 14), fg="#76787C",

                    bg="#f0e99a", relief="groove", justify="center")

    e\_grade.grid(row=7, column=7)

    # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

    show\_window = Frame(regis\_form\_window, bg="#C1D9F9")

    show\_window.pack(pady=5)

    # Add Some Style

    style = ttk.Style()

    # Pick a theme

    style.theme\_use("default")  # default,clam,alt

    # Configre our treeview colors

    style.configure("Treeview",

                    background="white",

                    forground="black",

                    rowheight=25,

                    fieldbackground="white")

    # Change Selected Color

    style.map("Treeview",

              background=[('selected', 'blue')])

    # Create treeview Frame

    tree\_frame = Frame(show\_window, bg="yellow")

    tree\_frame.pack(pady=10)

    # Treeview Scrollbar

    tree\_scroll = Scrollbar(tree\_frame)

    tree\_scroll.pack(side="right", fill=Y)

    # Create Treeview

    my\_tree = ttk.Treeview(tree\_frame, yscrollcommand=tree\_scroll.set)

    # pack to the screen

    my\_tree.pack()

    # Configure the scrollbar

    tree\_scroll.config(command=my\_tree.yview)

    # Define Our Columns

    my\_tree['columns'] = ("First Name", "Last Name",

                          "Student ID", "Gmail",

                          "Image Certificare", "DOB",

                          "Gender", "Grade",

                          "Phone Number", "University",

                          "College", "Department")

    # Format Our Columns

    my\_tree.column("#0", width=0, stretch=NO)

    my\_tree.column("First Name", anchor=CENTER, width=100)

    my\_tree.column("Last Name", anchor=CENTER, width=100)

    my\_tree.column("Student ID", anchor=CENTER, width=100)

    my\_tree.column("Gmail", anchor=CENTER, width=130)

    my\_tree.column("Image Certificare", anchor=CENTER, width=120)

    my\_tree.column("DOB", anchor=CENTER, width=100)

    my\_tree.column("Gender", anchor=CENTER, width=100)

    my\_tree.column("Grade", anchor=CENTER, width=100)

    my\_tree.column("Phone Number", anchor=CENTER, width=120)

    my\_tree.column("University", anchor=CENTER, width=100)

    my\_tree.column("College", anchor=CENTER, width=100)

    my\_tree.column("Department", anchor=CENTER, width=120)

    # Create Headings

    my\_tree.heading("#0", text="", anchor=CENTER)

    my\_tree.heading("First Name", text="First Name", anchor=CENTER)

    my\_tree.heading("Last Name", text="Last Name", anchor=CENTER)

    my\_tree.heading("Student ID", text="Student ID", anchor=CENTER)

    my\_tree.heading("Gmail", text="Gmail", anchor=CENTER)

    my\_tree.heading("Image Certificare",

                    text="Image Certificare", anchor=CENTER)

    my\_tree.heading("DOB", text="DOB", anchor=CENTER)

    my\_tree.heading("Gender", text="Gender", anchor=CENTER)

    my\_tree.heading("Grade", text="Grade", anchor=CENTER)

    my\_tree.heading("Phone Number", text="Phone Number", anchor=CENTER)

    my\_tree.heading("University", text="University", anchor=CENTER)

    my\_tree.heading("College", text="College", anchor=CENTER)

    my\_tree.heading("Department", text="Department", anchor=CENTER)

    # Add Data

    connection = pyodbc.connect('Driver={SQL Server};'

                                'Server=HOZANPC;'

                                'Database=userDB2;'

                                'Trusted\_Connection=yes;')

    cursorObject = connection.cursor()

    global data

    data = cursorObject.execute('''select \* from userDB2.dbo.University u

    join College c

    on u.Univ\_ID=c.Univ\_ID

    join Department d

    on c.College\_ID=d.College\_ID

    join Enrollment e

    on d.Dep\_ID=e.Dep\_ID

    join Student s

    on e.Student\_ID=s.Student\_ID;''')

    # Create striped row tags

    my\_tree.tag\_configure('oddRow', background="white")

    my\_tree.tag\_configure('evenRow', background="lightblue")

    global count

    count = 0

    for record in data:

        if count % 2 == 0:

            my\_tree.insert(parent='', index='end', iid=count, text="",

                           values=(record[18], record[19], record[17],

                                   record[22], record[14], record[21],

                                   record[20], record[13], record[23],

                                   record[1], record[4], record[8]),

                           tags=('evenRow'))

        else:

            my\_tree.insert(parent='', index='end', iid=count, text="",

                           values=(record[18], record[19], record[17],

                                   record[22], record[14], record[21],

                                   record[20], record[13], record[23],

                                   record[1], record[4], record[8]),

                           tags=('oddRow'))

        count += 1

    update\_ID = ''

    update\_mark = ''

    global row\_to\_list

    global values

    # Add Record

    # put values of Entry in variables for using it in database

    def add\_record():

        fname = e\_first\_name.get()

        lname = e\_last\_name.get()

        global st\_ID

        st\_ID = e\_id.get()

        gmail = e\_gmail.get()

        phone\_num = e\_phone.get()

        image = v\_image.get()

        DOB = v\_date.get()

        gender = va.get()

        mark = e\_grade.get()

        univ = univ\_combo.get()

        coll = coll\_combo.get()

        dep = dep\_combo.get()

        # new connection with database

        connection = pyodbc.connect('Driver={SQL Server};'

                                    'Server=HOZANPC;'

                                    'Database=userDB2;'

                                    'Trusted\_Connection=yes;')

        cursorObject = connection.cursor()

query=cursorObject.execute("select \* from userDB2.dbo.Student where Student\_ID=(?) or Email=(?) or Phone\_Number=(?)",

(st\_ID,gmail,phone\_num))

        if(fname == '' or lname == '' or st\_ID == '' or

         gmail == '' or phone\_num == '' or image == '' or

          DOB == '' or gender == ' ' or mark == '' or

           univ == '' or coll == '' or dep == ''):

            messagebox.showwarning(

                "Registration", "The Entries Must Not Be Empty All Of It Required",

                 parent=regis\_form\_window)

        elif(query.fetchall()):

            messagebox.showwarning(

                "Registration", " This Student ID or Gmail or Phone Number Is Already Exist",

                 parent=regis\_form\_window)

            e\_first\_name.delete(0, END)

            e\_last\_name.delete(0, END)

            e\_id.delete(0, END)

            e\_gmail.delete(0, END)

            e\_phone.delete(0, END)

            v\_image.set("")

            v\_date.set("")

            va.set(" ")

            e\_grade.delete(0, END)

            univ\_combo.delete(0, END)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

        else:

            messagebox.showinfo(

                "Registration", "New Student Added Successfully",

                 parent=regis\_form\_window)

            st\_ID = int(e\_id.get())

            enroll\_ID = st\_ID+10

            mark = float(e\_grade.get())

            data2 = cursorObject.execute('''select e.Dep\_ID, d.Dep\_ID from University u

            join College c

            on u.Univ\_ID=c.Univ\_ID

            join Department d

            on c.College\_ID=d.College\_ID

            join Enrollment e

            on d.Dep\_ID=e.Dep\_ID

            join Student s

            on e.Student\_ID=s.Student\_ID

            where u.Univ\_Name=(?) and c.College\_Name=(?) and d.Dep\_Name=(?); ''',

             (univ, coll, dep))

            global row\_to\_list

            row\_to\_list = [123]

            print(row\_to\_list)

            for row in data2:

                row\_to\_list = [elem for elem in row]

            print(row\_to\_list)

            dep\_ID = row\_to\_list[0]

            sql\_command\_stud = (

                '''insert into userDB2.dbo.Student values(?,?,?,?,?,?,?)''')

            sql\_command\_enroll = ('''insert into userDB2.dbo.Enrollment (Enrol\_ID,Enrollment\_Date,

            Grade,Image\_Certificate,Student\_ID,Dep\_ID)

            values(?,?,?,?,?,?)''')

            cursorObject.execute(sql\_command\_stud,

                                 (st\_ID, fname, lname, gender,

                                  DOB, gmail, phone\_num))

            Enroll\_date = str(date.today())

            cursorObject.execute(

                sql\_command\_enroll, (enroll\_ID, Enroll\_date, mark,

                 image, st\_ID, dep\_ID))

            connection.commit()

            connection.close()

            my\_tree.tag\_configure('oddRow', background="white")

            my\_tree.tag\_configure('evenRow', background="lightblue")

            global count

            if count % 2 == 0:

                my\_tree.insert(parent='', index='end', iid=count, text="",

                               values=(e\_first\_name.get(), e\_last\_name.get(), e\_id.get(),

                                       e\_gmail.get(), v\_image.get(), v\_date.get(),

                                       va.get(), e\_grade.get(), e\_phone.get(),

                                       univ\_combo.get(), coll\_combo.get(), dep\_combo.get()),

                               tags=('evenRow'))

            else:

                my\_tree.insert(parent='', index='end', iid=count, text="",

                               values=(e\_first\_name.get(), e\_last\_name.get(), e\_id.get(),

                                       e\_gmail.get(), v\_image.get(), v\_date.get(),

                                       va.get(), e\_grade.get(), e\_phone.get(),

                                       univ\_combo.get(), coll\_combo.get(), dep\_combo.get()),

                               tags=('oddRow'))

            count += 1

            # Cleare Entry Boxes

            e\_first\_name.delete(0, END)

            e\_last\_name.delete(0, END)

            e\_id.delete(0, END)

            e\_gmail.delete(0, END)

            e\_phone.delete(0, END)

            v\_image.set("")

            v\_date.set("")

            va.set(" ")

            e\_grade.delete(0, END)

            univ\_combo.delete(0, END)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

    def remove\_one():  # Remove the selected one

        selected = my\_tree.focus()

        if selected == '':

            messagebox.showerror(

                "Registration", "You Should To Select First",

                 parent=regis\_form\_window)

            return

        messagebox.showinfo(

            "Registration", "The Record Has Been Successfully Deleted",

             parent=regis\_form\_window)

        # my\_tree.selection()[0] for select one item only

        x = my\_tree.selection()[0]

        values = my\_tree.item(selected, 'values')  # Getting the whole row

        my\_tree.delete(x)

        # new connection with database

        connection = pyodbc.connect('Driver={SQL Server};'

                                    'Server=HOZANPC;'

                                    'Database=userDB2;'

                                    'Trusted\_Connection=yes;')

        cursorObject = connection.cursor()

        stu\_id = int(values[2])

        grade = float(values[7])

        cursorObject.execute(

            "delete from userDB2.dbo.Enrollment where Student\_ID=(?) and Grade=(?)",

             (stu\_id, grade))

        cursorObject.execute(

            "delete from userDB2.dbo.Student where Student\_ID=(?)",

             stu\_id)

        connection.commit()

        connection.close()

    def select\_record():  # Select Record

        # Cleare Entry Boxes

        e\_first\_name.delete(0, END)

        e\_last\_name.delete(0, END)

        e\_id.delete(0, END)

        e\_gmail.delete(0, END)

        e\_phone.delete(0, END)

        v\_image.set("")

        v\_date.set("")

        va.set(" ")

        e\_grade.delete(0, END)

        univ\_combo.delete(0, END)

        coll\_combo.delete(0, END)

        dep\_combo.delete(0, END)

        # grab record number

        selected = my\_tree.focus()

        if selected == '':

            messagebox.showerror(

                "Registration", "You Should To Select First",

                 parent=regis\_form\_window)

            return

        # grab record values

        global values

        values = my\_tree.item(selected, 'values')

        # print(values)

        # temp\_label.config(text=values[0])  #Return Tuple

        # Output to entry boxes

        e\_first\_name.insert(0, values[0])

        e\_last\_name.insert(0, values[1])

        global update\_ID

        update\_ID = values[2]

        e\_id.insert(0, values[2])

        e\_gmail.insert(0, values[3])

        v\_image.set(values[4])

        v\_date.set(values[5])

        va.set(values[6])

        global update\_mark

        update\_mark = values[7]

        e\_grade.insert(0, values[7])

        e\_phone.insert(0, values[8])

        univ\_combo.insert(0, values[9])

        coll\_combo.insert(0, values[10])

        dep\_combo.insert(0, values[11])

    def update\_record():  # Save Update Record

        global update\_ID

        global update\_mark

        # grab record number

        selected = my\_tree.focus()

        if selected == '':

            messagebox.showerror(

                "Registration", "You Should To Select First",

                 parent=regis\_form\_window)

            return

        # put values of Entry in variables for using it in database

        fname = e\_first\_name.get()

        lname = e\_last\_name.get()

        global st\_ID

        st\_ID = e\_id.get()

        gmail = e\_gmail.get()

        phone\_num = e\_phone.get()

        image = v\_image.get()

        DOB = v\_date.get()

        gender = va.get()

        mark = e\_grade.get()

        univ = univ\_combo.get()

        coll = coll\_combo.get()

        dep = dep\_combo.get()

        # new connection with database

        connection = pyodbc.connect('Driver={SQL Server};'

                                    'Server=HOZANPC;'

                                    'Database=userDB2;'

                                    'Trusted\_Connection=yes;')

        cursorObject = connection.cursor()

        # Delete Record

        cursorObject.execute("delete from userDB2.dbo.Enrollment where Student\_ID=(?) and Grade=(?)",

                             (update\_ID, update\_mark))

        cursorObject.execute(

            "delete from userDB2.dbo.Student where Student\_ID=(?)", update\_ID)

query=cursorObject.execute("select \* from userDB2.dbo.Student where Student\_ID=(?) or Email=(?) or Phone\_Number=(?)",

(st\_ID,gmail,phone\_num))

        if(fname == '' or lname == '' or st\_ID == '' or

         gmail == '' or phone\_num == '' or image == '' or

          DOB == '' or gender == ' ' or mark == '' or

           univ == '' or coll == '' or dep == ''):

            messagebox.showwarning(

                "Registration", "The Entries Must Not Be Empty All Of It Required",

                 parent=regis\_form\_window)

        elif(query.fetchall()):

            messagebox.showwarning(

                "Registration", " This Student ID or Gmail or Phone Number Is Already Exist",

                 parent=regis\_form\_window)

            e\_first\_name.delete(0, END)

            e\_last\_name.delete(0, END)

            e\_id.delete(0, END)

            e\_gmail.delete(0, END)

            e\_phone.delete(0, END)

            v\_image.set("")

            v\_date.set("")

            va.set(" ")

            e\_grade.delete(0, END)

            univ\_combo.delete(0, END)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

        else:

            messagebox.showinfo(

                "Registration", "The Student Updated Successfully",

                 parent=regis\_form\_window)

            st\_ID = int(e\_id.get())

            enroll\_ID = st\_ID+10

            mark = float(e\_grade.get())

            data2 = cursorObject.execute('''select e.Dep\_ID, d.Dep\_ID from University u

            join College c

            on u.Univ\_ID=c.Univ\_ID

            join Department d

            on c.College\_ID=d.College\_ID

            join Enrollment e

            on d.Dep\_ID=e.Dep\_ID

            join Student s

            on e.Student\_ID=s.Student\_ID

            where u.Univ\_Name=(?) and c.College\_Name=(?) and d.Dep\_Name=(?); ''',

             (univ, coll, dep))

            global row\_to\_list

            row\_to\_list = [123]

            for row in data2:

                row\_to\_list = [elem for elem in row]

            print(row\_to\_list)

            dep\_ID = row\_to\_list[0]

            # ADD Record

            sql\_command\_stud = (

                '''insert into userDB2.dbo.Student values(?,?,?,?,?,?,?)''')

            sql\_command\_enroll = ('''insert into userDB2.dbo.Enrollment (Enrol\_ID,Enrollment\_Date,

            Grade,Image\_Certificate,Student\_ID,Dep\_ID)

            values(?,?,?,?,?,?)''')

            cursorObject.execute(sql\_command\_stud,

                                 (st\_ID, fname, lname, gender,

                                  DOB, gmail, phone\_num))

            Enroll\_date = str(date.today())

            cursorObject.execute(

                sql\_command\_enroll, (enroll\_ID, Enroll\_date,

                 mark, image, st\_ID, dep\_ID))

            connection.commit()

            connection.close()

            # save new data

            my\_tree.item(selected, text="",

                         values=(e\_first\_name.get(), e\_last\_name.get(), e\_id.get(),

                                 e\_gmail.get(), v\_image.get(), v\_date.get(),

                                 va.get(), e\_grade.get(), e\_phone.get(),

                                 univ\_combo.get(), coll\_combo.get(), dep\_combo.get(),))

            # Clear Entry Boxes

            e\_first\_name.delete(0, END)

            e\_last\_name.delete(0, END)

            e\_id.delete(0, END)

            e\_gmail.delete(0, END)

            e\_phone.delete(0, END)

            v\_image.set("")

            v\_date.set("")

            va.set(" ")

            e\_grade.delete(0, END)

            univ\_combo.delete(0, END)

            coll\_combo.delete(0, END)

            dep\_combo.delete(0, END)

    def reset\_record():

        # Clear Entry Boxes

        e\_first\_name.delete(0, END)

        e\_last\_name.delete(0, END)

        e\_id.delete(0, END)

        e\_gmail.delete(0, END)

        e\_phone.delete(0, END)

        v\_image.set("")

        v\_date.set("")

        va.set(" ")

        e\_grade.delete(0, END)

        univ\_combo.delete(0, END)

        coll\_combo.delete(0, END)

        dep\_combo.delete(0, END)

    # Buttons

    reset\_button = Button(show\_window, text="Reset Record",

     cursor="hand2",bg="#33AD6A", fg="white",

      font=('Century', 12), relief="flat",

       overrelief="ridge", command=reset\_record)

    reset\_button.pack(side="left", padx=80)

    select\_button = Button(show\_window, text="Select Record",

     cursor="hand2",bg="#33AD6A", fg="white",

      font=('Century', 12), relief="flat",

      overrelief="ridge", command=select\_record)

    select\_button.pack(side="left", padx=60)

    update\_button = Button(show\_window, text="Save Record",

     cursor="hand2",bg="#33AD6A", fg="white",

      font=('Century', 12), relief="flat",

      overrelief="ridge", command=update\_record)

    update\_button.pack(side="left", padx=60)

    add\_records = Button(show\_window, text="Add Record",

     cursor="hand2",bg="#33AD6A", fg="white",

      font=('Century', 12), relief="flat", overrelief="ridge",

       command=add\_record)

    add\_records.pack(side="left", padx=60)

    # Remove the selected one

    remove\_one = Button(show\_window,

     text="Delete One Selected",

      cursor="hand2",bg="#33AD6A", fg="white",

      font=('Century', 12), relief="flat",

       overrelief="ridge", command=remove\_one)

    remove\_one.pack(side="left", padx=80, pady=10)

    connection.commit()

    connection.close()

    regis\_form\_window.mainloop()

def submit2():

    user = e1.get()

    password = e2.get()

    connection = pyodbc.connect('Driver={SQL Server};'

                                'Server=HOZANPC;'

                                'Database=userDB2;'

                                'Trusted\_Connection=yes;')

    cursorObject = connection.cursor()

    query = cursorObject.execute(

        "select \* from userDB2.dbo.users where userName=(?) and password=(?)",

         (user, password))

    if(query.fetchall()):

        messagebox.showinfo("Login", "Login Successfully",

         parent=root2)

        dashBoard()

    elif user == '' or password == '':

        messagebox.showwarning(

            "Login", "The Entry Must Not Be Empty",

             parent=root2)

    else:

        messagebox.showerror(

            "Login", "Wrong Username or Password",

             parent=root2)

        e1.delete(0, END)

        e2.delete(0, END)

    connection.commit()

    connection.close()



def login():

    global e1, e2, root2

    root2 = Toplevel(root)

    root2.title('Login Window')

    root2.focus\_force()

    root2.resizable(False, False)

    root2.geometry('+250+10')

    root2.iconbitmap(r'student icon.ico')

    root2.config(bg='white')

    f1 = Frame(root2, bg="white")

    f1.pack(side='top')

    f2 = Frame(root2, bg="white")

    f2.pack(side='top', pady=16)

    f3 = Frame(root2, bg="white")

    f3.pack(side='left', pady=16)

    Label(f1, text='Student Enrollment System', font=(

        'Century', 20, 'bold'), bg="white").pack(pady=26)

    p1 = PhotoImage(file='login Enrollment.png')

    Label(f1, image=p1, bg="white").pack()

    Label(f1, text='Login Form', font=(

        'Century', 20, 'bold'), bg="white").pack(pady=16)

    Label(f1, text='User Name: ',

     font=('Century', 18, 'bold'),

          bg="white").pack(side='left', padx=30)

    e1 = Entry(f1, font=('Arial', 16, 'bold'),

     fg="#76787C", bg='#ccc')

    e1.pack(side='left')

    e1.focus\_force()

    Label(f2, text='Password: ',

     font=('Century', 18, 'bold'),

          bg="white").pack(side='left', padx=14)

    e2 = Entry(f2, font=('Arial', 16, 'bold'),

               fg="#76787C", bg='#ccc', show='\*')

    e2.pack(side='left')

    cb = PhotoImage(file="cancelB2.png")

    sb = PhotoImage(file="checked2.png")

    Button(f3, image=cb, bg="white",

     relief=FLAT, takefocus=1,

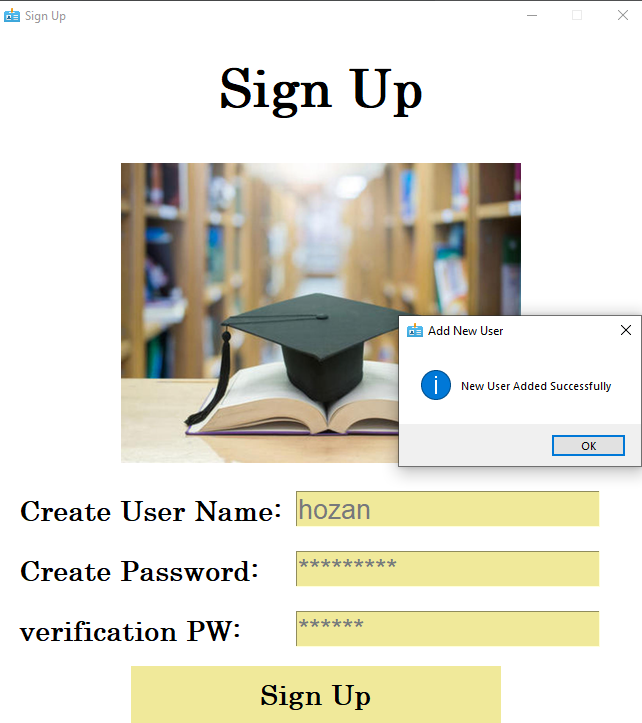
      command=lambda: root2.destroy(),cursor="hand2",

            overrelief='ridge').pack(side='left', padx=160)

    Button(f3, image=sb, bg="white", relief=FLAT,

     takefocus=2, command=submit2,cursor="hand2",

      overrelief='ridge').pack(side='left')

    root2.mainloop()

def submit():

    new\_user = v1.get()

    new\_password = v2.get()

    conf = v3.get()

    if new\_user == '' or new\_password == '' or conf == '':

        messagebox.showwarning(

            "Login", "The Entry Must Not Be Empty",

             parent=sign\_up\_window)

        return

    if conf == '123456':  # verifacation code

        connection = pyodbc.connect('Driver={SQL Server};'

                                    'Server=HOZANPC;'

                                    'Database=userDB2;'

                                    'Trusted\_Connection=yes;')

        cursorObject = connection.cursor()

        query = cursorObject.execute("select \* from userDB2.dbo.users where userName=(?) and password=(?)",

                                     (new\_user, new\_password))

        if(query.fetchall()):

            messagebox.showwarning(

                "Sign UP", "User Name Exist",

                 parent=sign\_up\_window)

            v1.set('')

            v2.set('')

            v3.set('')

        else:

            query = cursorObject.execute("INSERT INTO userDB2.dbo.users(userName,password) values(?,?)",

                                         (new\_user, new\_password))

            messagebox.showinfo(

                "Add New User", "New User Added Successfully",

                 parent=sign\_up\_window)

            v1.set('')

            v2.set('')

            v3.set('')

        sign\_up\_window.destroy()

        connection.commit()

        connection.close()

    else:

        messagebox.showerror(

            "Wrong Sign Up", "You are not allowed to create admin account",

             parent=sign\_up\_window)

        v1.set('')

        v2.set('')

        v3.set('')

def sign\_up():

    global v1

    global v2

    global v3

    global sign\_up\_window

    sign\_up\_window = Toplevel(root)

    sign\_up\_window.geometry("650x700+500+0")

    sign\_up\_window.title("Sign Up")

    sign\_up\_window.iconbitmap(r'student icon.ico')

    sign\_up\_window.focus\_force()

    sign\_up\_window.config(bg="white")

    sign\_up\_window.resizable(False, False)

    Label(sign\_up\_window, text="Sign Up",

     font=('Century', 40, 'bold'),

          bg="white").pack(anchor=N,

           padx=10, pady=20)

    Label(sign\_up\_window, image=top\_pic,

     bg="white").pack(pady=20)

    l\_creat\_username = Label(sign\_up\_window,

     text="Create User Name:   ", font=(

        'Century', 20, 'bold'),

         bg="white").place(x=20, y=460)

    v1 = StringVar('')

    e\_username = Entry(sign\_up\_window,

     textvariable=v1, font=('Arial', 20),

      fg="#76787C", bg="#f0e99a")

    e\_username.place(x=300, y=460)

    e\_username.focus\_force()

    l\_password = Label(sign\_up\_window,

     text="Create Password:   ", font=(

        'Century', 20, 'bold'),

         bg="white").place(x=20, y=520)

    v2 = StringVar('')

    e\_password = Entry(sign\_up\_window,

     textvariable=v2, font=('Arial', 20),

      fg="#76787C",bg="#f0e99a",

       show="\*").place(x=300, y=520)

    l\_ver = Label(sign\_up\_window,

     text="verification PW:   ", font=(

        'Century', 20, 'bold'),

         bg="white").place(x=20, y=580)

    v3 = StringVar('')

    e\_ver = Entry(sign\_up\_window,

     textvariable=v3, font=(

        'Arial', 20), fg="#76787C",

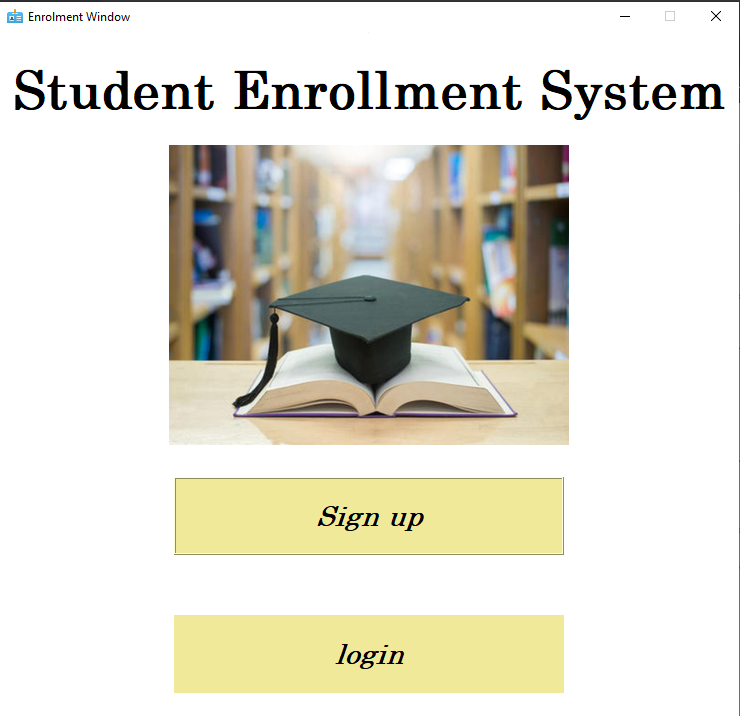
        bg="#f0e99a", show="\*").place(x=300, y=580)

    sub\_button = Button(sign\_up\_window,

    text="Sign Up", font=('Century', 20, 'bold'),command=submit,

    bg="#f0e99a", relief="flat",

    overrelief="ridge", width=20).place(x=135, y=635)



root = Tk()

root.geometry("+600+0")

root.title("Enrolment Window ")

root.config(bg="white")

root.resizable(False, False)

root.iconbitmap(r'student icon.ico')

root.focus\_force()

# =======frame

top\_frame = Frame(root).pack(side=TOP)

mid\_frame = Frame(root).pack(side=BOTTOM)

bootom\_frsme = Frame(root)

lab1 = Label(top\_frame,

 text="Student Enrollment System", font=(

    'Century', 40, 'bold'),

     bg="white").pack(anchor=N, padx=10, pady=20)

top\_pic = PhotoImage(file="university enroll.png")

label\_pic = Label(top\_frame, image=top\_pic,

 bg="white").pack(anchor=CENTER)

# ====================middle

signup\_button = Button(mid\_frame, text="Sign up",

 width="20", bg="#f0e99a", command=sign\_up,

 font=('Century', 20, 'italic', 'bold'),

  relief="flat", overrelief="ridge")

signup\_button.pack(padx=30, pady=30, ipadx=10, ipady=10)

login\_button = Button(mid\_frame,

 text="login", width="20", bg="#f0e99a",

  font=('Century', 20, 'italic', 'bold'),

   relief="flat", overrelief="ridge", command=login)

login\_button.pack(padx=30, pady=30, ipadx=10, ipady=10)

root.mainloop()