



# POR T CITY INTERNATIONAL UNIVERSITY

## Final Term Report

**Course Code: CSE 434**

**Course Title: Web Technology Sessional.**

**Date Of Submission: 18 August 2025**

### Submitted To:

**Student's Name: Faria Jahan Janie.**

**Program: BSc in CSE**

**Batch: 28th (A)**

**ID: CSE 028 07551**

**Port City International University**

### Submitted To:

**Name of lecturer**

**Subashis Roy Bhowmik**

**Department: CSE**

**Port City International University**

**Project Name:** Luvia Medical Center

**With Smart AI Diagnosis Hub**

**Project Subtitle:** Hospital Management System with AI powered.

## **Table of Contents**

<b>1. Introduction</b>	<b>03</b>
• Introduction	
• Motivation	
• Objectives	
<b>2. System Technologies Used</b>	<b>04</b>
• Technologies	
• System Requirements	
• Application Structure	
<b>3. Source Codes</b>	<b>07</b>
• Admin Panel	
• User Panel	
<b>4. Output</b>	<b>17</b>
<b>5. Conclusion</b>	<b>31</b>

## 1. Introduction

---

### **Introduction:**

Luvia Medical Center is an innovative, web-based hospital management system designed to simplify healthcare operations and provide intelligent health predictions. Built on the Flask framework, it integrates machine learning models for Diabetes and Heart Disease prediction, a MySQL database for secure data storage, and Power BI dashboards for advanced data visualization. The system caters to patients, doctors, and administrators, ensuring smooth operations, better decision-making, and improved patient care. With its user-friendly interface, role-based access, and predictive healthcare features, Luvia Medical Center delivers a modern, reliable, and comprehensive solution for hospital management.

### **Motivation:**

In today's fast-paced world, people are constantly seeking smarter, faster, and more reliable solutions for managing their health and daily needs. Traditional approaches often lack real-time insights, proper data-driven decision-making, and personalized services. With the rise of Machine Learning and Data Visualization tools like Power BI, we now have the opportunity to turn raw data into meaningful predictions and insights. By combining these technologies with the simplicity and flexibility of Flask, this project is motivated to:

- Bridge the gap between raw medical data and actionable predictions.
- Provide users with easy-to-use, interactive web applications.
- Enable better healthcare decisions through predictive ML models.
- Present insights in a visually engaging way using dashboards.

This project is inspired by the vision of creating a smart, accessible, and data-driven medical platform that can positively impact people's lives.

### **Objective:**

The main objectives of this project are:

**Donor Management** – Enable easy registration, editing, and management of donor profiles.

**Disease Prediction** – Build ML-powered prediction models (Diabetes, Heart, Pregnancy, etc.) integrated with Flask.

**User-Friendly Interface** – Ensure patients and admins can easily interact through a clean, responsive design.

**Secure Data Handling** – Implement proper authentication, database management, and privacy control.

Interactive Analytics – Use Power BI and dashboards to visualize data dynamically.

**Centralized Management** – Create an all-in-one platform for doctors, donors, patients, and admins to manage healthcare efficiently.

**Scalability** – Develop the system in a way that can expand with more ML models and features in the future.

## 2. System Technologies Used

---

### Technologies:

#### 1. Programming Language

- **Python 3.10+** – Backend development and ML model integration

#### 2. Web Framework

- **Flask** – Lightweight Python web framework for building the web application

#### 3. Frontend Technologies

- **HTML5** – Structuring web pages
- **CSS3 / Bootstrap / Tailwind CSS** – Styling and responsive design
- **JavaScript** – Interactive UI components

#### 4. Database

- **SQLite / MySQL** – Storing user data, hospital records, and form submissions

## 5. Libraries & Packages

- **Pandas, NumPy** – Data processing for ML models
- **Scikit-learn** – Machine learning model implementation
- **Flask-WTF** – Form validation
- **Flask-Login** – User authentication
- **Jinja2** – HTML templating

## 6. Visualization & BI Tools

- **Power BI** – Advanced data visualization and dashboard reporting for insights

## 7. Development Tools

- **VS Code** – IDE for coding
- **Git & GitHub** – Version control and repository management

## 8. Operating System

- Windows 11 pro

### Run the Application

```
flask run
```

Open your browser at: <http://127.0.0.1:5000/>

# System Requirements

### Hardware Requirements

- **Processor:** Intel i3
- **RAM:** Minimum 4 GB (8 GB recommended)
- **Storage:** Minimum 2 GB free (5 GB recommended)
- **Display:** 1366x768 resolution or higher

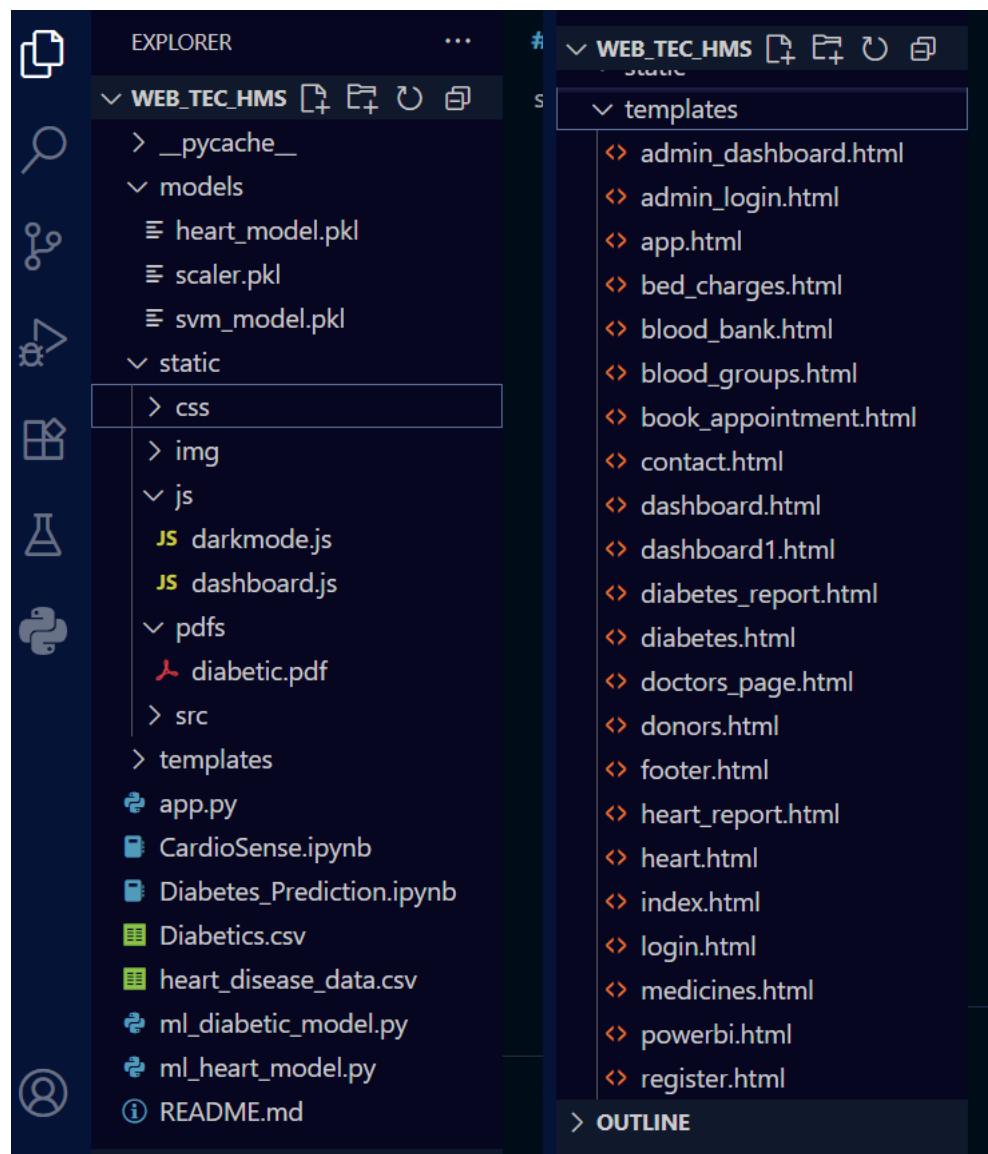
### Software Requirements

- **Operating System:** Windows 10/11

- **Python:** Version 3.8+ (3.10 recommended)
- **Database:** MySQL Server 8.0+
- **Web Browser:** Chrome, Firefox, or Edge (latest versions)
- **Libraries/Frameworks:** Flask, Pandas, NumPy, Scikit-Learn
- **Frontend:** HTML5, CSS3, JavaScript (Tailwind CSS optional)

## Application Structure:

Structure of my project files:



### 3. Source Code

- App.py file:

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Back, Forward, Home, Refresh, Stop, Run, Cell, Kernel, Help.
- Search Bar:** WEB\_TEC\_HMS
- Left Sidebar (EXPLORER):**
  - WEB\_TEC\_HMS
  - templates
    - dashboard1.html
    - diabetes\_report.html
    - diabetes.html
    - doctors.page.html
    - donors.html
    - footer.html
    - heart\_report.html
    - heart.html
    - index.html
    - login.html
    - medicines.html
    - powerbi.html
    - register.html
  - app.py
  - CardioSense.ipynb
  - Diabetes\_Prediction.ipynb
  - Diabetics.csv
  - heart\_disease\_data.csv
  - ml.diabetic\_model.py
  - ml.heart\_model.py
- Current File:** app.py
- Code Content:** The code defines a Flask application for diabetes prediction. It imports various modules from Flask, wtforms, and MySQLdb. It configures MySQL settings and creates an app object. It also imports functions for predicting heart disease and diabetes from ml modules. A Register Form is defined at the bottom.

## Install Libraries in Flask and add MySQL

```
  ● scaler.pkl          23
  ● svm_model.pkl      24  # ----- Register Form -----
  ✓ static              25  class RegisterForm(FlaskForm):
  > css                 26      name = StringField("Name", validators=[DataRequired()])
  > img                 27      email = StringField("Email", validators=[DataRequired(), Email()])
  > js                  28      password = PasswordField("Password", validators=[DataRequired()])
  ✓ js                  29      submit = SubmitField("Register")
  > darkmode.js        30
  > dashboard.js       31
  ✓ pdfs               32  def validate_email(self, field):
  > diabetic.pdf       33      cursor = mysql.connection.cursor()
  ✓ src                34      cursor.execute("SELECT * FROM users WHERE email = %s", (field.data,))
  > templates          35      user = cursor.fetchone()
  > admin_dashboard.html 36      cursor.close()
  > admin_login.html   37      if user:
  > app.html           38          raise ValidationError("Email already registered.")
  > bed_charges.html  39
  > blood_bank.html   40  # ----- Login Form -----
  > blood_groups.html 41  class LoginForm(FlaskForm):
  > book_appointment.html 42      email = StringField("Email", validators=[DataRequired(), Email()])
  > contact.html       43      password = PasswordField("Password", validators=[DataRequired()])
  > dashboard.html     44      submit = SubmitField("Login")
  > diabetes_report.html 45
  > diabetes.html      46  # ----- Home Page -----
  > doctors_page.html 47  @app.route('/')
  > donors.html        48  def index():
  > footer.html        49      return render_template('index.html')
  > heart_report.html  50
  > heart.html          51  # ----- Register -----
  > index.html          52  @app.route('/register', methods=['GET', 'POST'])
  > login.html          53  def register():
  > medicines.html     54      form = RegisterForm()
  > powerbi.html       55      if form.validate_on_submit():
  > register.html      56          name = form.name.data
  > app.py             57          email = form.email.data
  > CardioSense.ipynb 58          password = form.password.data
  > Diabetes_Prediction.ipynb 59          hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
  > Diabetes.csv        60
  > heart_disease_data.csv 61          try:
  > ml_diabetic_model.py 62              cursor = mysql.connection.cursor()
  > ml_heart_model.py   63              cursor.execute("INSERT INTO users (name, email, password) VALUES (%s, %s, %s)", (name, email, hashed_password))
  > READMEmd            64              mysql.connection.commit()
  > READMEmd            65              cursor.close()
  > READMEmd            66              flash("Registration successful! Please log in.", "success")
  > READMEmd            67              return redirect(url_for('login'))
  > READMEmd            68          except Exception as e:
  > READMEmd            69              flash(f"Database Error: {str(e)}", "danger")
  > READMEmd            70
  > READMEmd            71  return render_template('register.html', form=form)
```

## Register and Login

```

    < models
      < heart_model.pkl
      < scaler.pkl
      < svm_model.pkl
    < static
      > css
      > img
    < js
      JS darkmode.js
      JS dashboard.js
    < pdfs
      ↳ diabetic.pdf
    < src
    < templates
      ◇ admin_dashboard.html
      ◇ admin_login.html
      ◇ app.html
      ◇ bed_charges.html
      ◇ blood_bank.html
      ◇ blood_groups.html
      ◇ book_appointment.html
      ◇ contact.html
      ◇ dashboard.html
      ◇ dashboard1.html
      ◇ diabetes_report.html
  146
  147 @app.route('/diabetes')
  148 def diabetes():
  149     |   return render_template('diabetes.html')
  150
  151
  152 @app.route('/predict_diabetes', methods=['POST'])
  153 def predict_diabetes_route():
  154     # Extract data from form inputs
  155     data = [
  156         float(request.form['pregnancies']),
  157         float(request.form['glucose']),
  158         float(request.form['bloodpressure']),
  159         float(request.form['skinthickness']),
  160         float(request.form['insulin']),
  161         float(request.form['bmi']),
  162         float(request.form['dpf']),
  163         float(request.form['age'])
  164     ]
  165
  166     # Use the prediction function from ml_diabetic_model.py
  167     prediction = predict_diabetes(data)
  168
  169     # Prepare user-friendly message
  170     result = "The person is Diabetic" if prediction == 1 else "The person is Not Diabetic"
  171
  172     return render_template('diabetes.html', prediction_text=result)
  173

```

## Diabetes Prediction

```

> __pycache__
< models
  < heart_model.pkl
  < scaler.pkl
  < svm_model.pkl
< static
  > css
  > img
  < js
    JS darkmode.js
    JS dashboard.js
  < pdfs
    ↳ diabetic.pdf
  < src
  < templates
    ◇ app.py
    ◇ CardioSense.ipynb
    ◇ Diabetes_Prediction.ipynb
    ◇ Diabetics.csv
    ◇ heart_disease.data.csv
    ◇ ml_diabetic_model.py
    ◇ ml_heart_model.py
    ◇ README.md
  ◇ OUTLINE
  ◇ TIMELINE
  ◇ A ...
  191
  192 # ❸ Route to show heart disease form
  193 @app.route('/heart')
  194 def heart():
  195     |   return render_template('heart.html')
  196
  197 # ❹ Route to show heart disease report
  198 @app.route('/predict_heart', methods=['POST'])
  199 def predict_heart_report():
  200     # Extract form data
  201     input_data = [
  202         float(request.form['age']),
  203         float(request.form['sex']),
  204         float(request.form['cp']),
  205         float(request.form['trestbps']),
  206         float(request.form['chol']),
  207         float(request.form['fbs']),
  208         float(request.form['restecg']),
  209         float(request.form['thalach']),
  210         float(request.form['exang']),
  211         float(request.form['oldpeak']),
  212         float(request.form['slope']),
  213         float(request.form['ca']),
  214         float(request.form['thal'])
  215     ]
  216     # Make prediction
  217     prediction = predict_heart(input_data)
  218     result_text = "The person has Heart Disease" if prediction == 1 else "The person is Healthy"
  219
  220     # Prepare data for the report template
  221     report_data = {
  222         'age': request.form['age'],
  223         'sex': request.form['sex'],
  224         'cp': request.form['cp'],
  225         'trestbps': request.form['trestbps'],
  226         'chol': request.form['chol'],
  227         'fbs': request.form['fbs'],
  228         'restecg': request.form['restecg'],
  229         'thalach': request.form['thalach'],
  230         'exang': request.form['exang'],
  231         'oldpeak': request.form['oldpeak'],
  232         'slope': request.form['slope'],
  233         'ca': request.form['ca'],
  234         'thal': request.form['thal'],
  235         'result': result_text
  236     }
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1298
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1699
  1700
  1701
  1702
  1703
  1704
  1705
  1706
  1707
  1708
  1709
  1710
  1711
  1712
  1713
  1714
  1715
  1716
  1717
  1718
  1719
  1720
  1721
  1722
  1723
  1724
  1725
  1726
  1727
  1728
  1729
  1730
  1731
  1732
  1733
  1734
  1735
  1736
  1737
  1738
  1739
  1740
  1741
  1742
  1743
  1744
  1745
  1746
  1
```

```

244
245     @app.route('/book_appointment', methods=['GET', 'POST'])
246     def book_appointment():
247         if request.method == 'POST':
248             name = request.form.get('name')
249             email = request.form.get('email')
250             phone = request.form.get('phone')
251             date = request.form.get('date')
252             time = request.form.get('time')
253             doctor = request.form.get('doctor')
254             message = request.form.get('message')
255
256             if not (name and email and date and time and doctor):
257                 flash("Please fill in all required fields.", "danger")
258                 return redirect(url_for('book_appointment'))
259
260             try:
261                 cursor = mysql.connection.cursor()
262                 insert_query = """
263                     INSERT INTO appointments (name, email, phone, date, time, doctor, message)
264                     VALUES (%s, %s, %s, %s, %s, %s, %s)
265                 """
266
267                 cursor.execute(insert_query, (name, email, phone, date, time, doctor, message))
268                 mysql.connection.commit()
269                 cursor.close()
270                 flash("Appointment successfully booked!", "success")
271                 return redirect(url_for('Login'))
272             except Exception as e:
273                 flash(f"Database error: {str(e)}", "danger")
274                 return redirect(url_for('book_appointment'))
275
276             return render_template('book_appointment.html')

```

## Book Appointment to consult with Doctor

```

277
278     @app.route('/blood_bank', methods=['GET', 'POST'])
279     def blood_bank():
280         if request.method == 'POST':
281             donor_name = request.form['donor_name']
282             email = request.form['email']
283             phone = request.form['phone']
284             blood_group = request.form['blood_group']
285             last_donation = request.form.get('last_donation')
286             health_condition = request.form.get('health_condition')
287
288             # Save this info to your database here
289
290             flash('Thank you for your donation registration!', 'success')
291             return redirect(url_for('blood_bank'))
292
293         return render_template('blood_bank.html')
294
295     @app.route('/donors', methods=['GET', 'POST'])
296     def donors():
297         cursor = mysql.connection.cursor()
298
299         if request.method == 'POST':
300             donor_name = request.form['donor_name']
301             email = request.form['email']
302             phone = request.form['phone']
303             blood_group = request.form['blood_group']
304             last_donation = request.form.get('last_donation')
305             health_condition = request.form.get('health_condition')
306
307             cursor.execute("""
308                 INSERT INTO donors (donor_name, email, phone, blood_group, last_donation, health_condition)
309                 VALUES (%s, %s, %s, %s, %s, %s)
310             """, (donor_name, email, phone, blood_group, last_donation, health_condition))
311             mysql.connection.commit()
312
313             flash('Congratulations! You have successfully joined as a donor.', 'success')
314
315             cursor.execute("SELECT * FROM donors ORDER BY created_at DESC")
316             donor_list = cursor.fetchall()
317             cursor.close()
318
319             return render_template('donors.html', donors=donor_list)
320

```

## Blood Section and Add new Donor

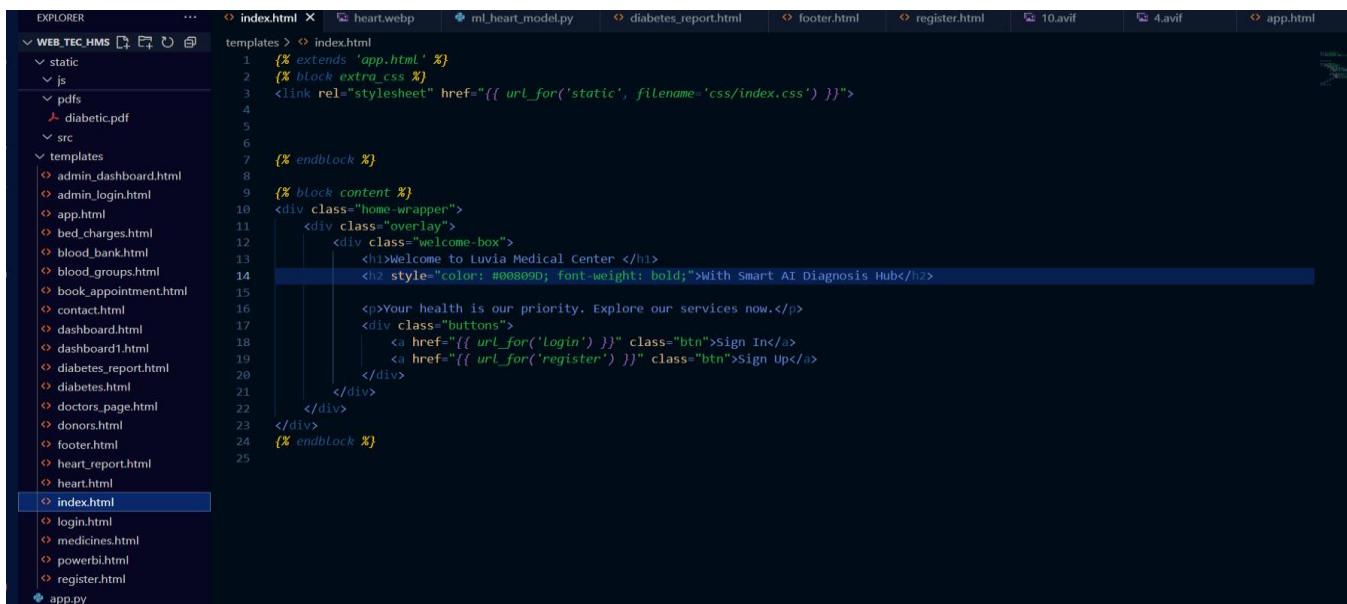
```

> img
  387 @app.route('/admin/dashboard')
  388 def admin_dashboard():
  389     if 'admin_id' not in session:
  390         flash('Please log in to access the admin dashboard.', "warning")
  391         return redirect(url_for('admin_login'))
  392
  393     cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
  394
  395     # Fetch users
  396     cursor.execute("SELECT id, name, email FROM users")
  397     users = cursor.fetchall()
  398
  399     # Fetch appointments
  400     cursor.execute("SELECT id, name, email, phone, date, time, doctor FROM appointments")
  401     appointments = cursor.fetchall()
  402
  403     # Fetch donors
  404     cursor.execute("SELECT id, donor_name, email, phone, blood_group FROM donors")
  405     donors = cursor.fetchall()
  406
  407     # Fetch admin
  408     cursor.execute("SELECT id, username, email, created_at FROM admin")
  409     admins = cursor.fetchall()
  410
  411     # Fetch contact messages
  412     cursor.execute("SELECT id, name, email, message, submitted_at FROM contact_messages")
  413     contact_messages = cursor.fetchall()
  414
  415     # Fetch doctors
  416     cursor.execute("SELECT id, name, email, phone, specialty, image FROM doctors")
  417
  418     doctors = cursor.fetchall()
  419
  420     cursor.close()
  421
  422     return render_template('admin_dashboard.html',
  423                           users=users,
  424                           appointments=appointments,
  425                           donors=donors,
  426                           admins=admins,
  427                           contact_messages=contact_messages,
  428                           doctors=doctors, # pass doctors to template
  429                           admin_email=session['admin_email'])
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451

```

## Admin Dashboard

- Admin & User Panel(html files)



The screenshot shows the Visual Studio Code interface with the Explorer sidebar open, displaying the project structure. The project includes static files, PDFs, and source code for templates like admin\_dashboard.html, admin\_login.html, and index.html. The index.html file is currently selected and its content is visible in the main editor area.

```

EXPLORER      ...
index.html ✘  heart.webp  ml_heart.model.py  diabetes_report.html  footer.html  register.html  10.avif  4.avif  app.html
templates > < index.html
1  {% extends "app.html" %} 
2  {% block extra_css %} 
3    <link rel="stylesheet" href="{{ url_for('static', filename='css/index.css') }}"/>
4
5
6
7  {% endblock %}
8
9  {% block content %}
10 <div class="home-wrapper">
11   <div class="overlay">
12     <div class="welcome-box">
13       <h1>Welcome to Luvia Medical Center </h1>
14       <h2 style="color: #008000; font-weight: bold;">With Smart AI Diagnosis Hub</h2>
15
16       <p>Your health is our priority. Explore our services now.</p>
17       <div class="buttons">
18         <a href="{{ url_for('Login') }}" class="btn">Sign In</a>
19         <a href="{{ url_for('register') }}" class="btn">Sign Up</a>
20       </div>
21     </div>
22   </div>
23 </div>
24 {% endBlock %}
25

```

## Index Page

EXPLORER    ... tes\_report.html    footer.html    register.html    10.avif    4.avif    app.html    admin\_dashboard.html    d ...

WEB.TEC.HMS

- models
  - scaler.pkl
  - svm\_model.pkl
- static
  - css
  - img
- js
  - darkmode.js
  - dashboard.js
- pdfs
  - diabetic.pdf
- src
- templates
  - admin\_dashboard.html
  - admin\_login.html
  - app.html
  - bed\_charges.html
  - blood\_bank.html
  - blood\_groups.html
  - book\_appointment.html
  - contact.html
  - dashboard.html
  - dashboard1.html
  - diabetes\_report.html

OUTLINE    TIMELINE

0 ▲ 0

```

1  {% extends 'app.html' %}

2
3  {% block content %}
4    <link rel="stylesheet" href="{{ url_for('static', filename='css/admin_dashboard.css') }}">
5
6    <div class="container">
7      <!-- Sidebar -->
8      <nav class="sidebar" id="sidebar">
9        <h3>Admin Menu</h3>
10       <a onclick="showSection('admins')" id="link-admins"><span class="icon">👤</span> Admin List</a>
11       <a onclick="showSection('doctors')" id="link-doctors"><span class="icon">👤</span> Doctors</a>
12       <a onclick="showSection('appointments')" id="link-appointments"><span class="icon">📅</span> Appointment List</a>
13       <a onclick="showSection('users')" id="link-users"><span class="icon">👤</span> User List</a>
14       <a onclick="showSection('donors')" id="link-donors"><span class="icon">📌</span> Donor List</a>
15       <a onclick="showSection('contact_messages')" id="link-contact_messages"><span class="icon">💬</span> Contact Messages</a>
16       <a onclick="togglePowerBIMenu()" id="link-powerbi"><span class="icon">📊</span> Power BI</a>
17       <div id="powerbi-submenu" style="display:none; padding-left: 20px;">
18         <a onclick="loadPowerBIPDF('diabetic.pdf')">Diabetes</a>
19         <a onclick="loadPowerBIPDF('cancer.pdf')">Cancer</a>
20     </div>
21   </nav>
22   <!-- Main Content -->
23   <section class="main-content">
24     <h1>Welcome, {{ admin_email }} <span class="icon">👤</span></h1>
25     <hr>
26
27
28
29
30
31
32

```

Ln 23, Col 3   Spaces: 2   UTF-8   CRLF   {} Django HTML   Go Live

## Admin Dashboard

WEB.TEC.HMS

- static
- js
- pdfs
  - diabetic.pdf
- src
- templates
  - admin\_dashboard.html
  - admin\_login.html
  - app.html
  - bed\_charges.html
  - blood\_bank.html
  - blood\_groups.html
  - book\_appointment.html
  - contact.html
  - dashboard.html
  - dashboard1.html
  - diabetes\_report.html
  - doctors.html
  - donors.html
  - footer.html
  - heart\_report.html
  - heart.html
  - index.html
  - login.html
  - medicines.html
  - powerbi.html
  - register.html

doctors.page.html

```

1  {% extends 'app.html' %}

2
3  {% block content %}
4    <link rel="stylesheet" href="{{ url_for('static', filename='css/doctors.css') }}">
5
6    <div class="container mt-5">
7      <h1 class="mb-5 text-center" style="font-weight: 700; color: #2c3e50;">Meet Our Doctors</h1>
8
9      <div class="row">
10        {% for doc in doctors %}
11          <div class="col-md-4 mb-4">
12            <div class="card doctor-card h-100 text-center p-3 border-0 shadow-sm rounded-4">
13              {% if doc.image %}
14                
15              {% else %}
16                
17              {% endif %}
18
19              <div class="card-body">
20                <h5 class="card-title mb-2" style="font-weight:600; color:#34495e;">{{ doc.name }}</h5>
21                <p class="card-text mb-1"><i class="bi bi-envelope-fill me-2"></i>{{ doc.email }}</p>
22                <p class="card-text mb-1"><i class="bi bi-telephone-fill me-2"></i>{{ doc.phone }}</p>
23                <p class="card-text mb-0"><i class="bi bi-heart-pulse-fill me-2"></i>{{ doc.specialty }}</p>
24              </div>
25            </div>
26          {% endfor %}
27        </div>
28
29
30
31
32

```

## Doctor Page

WEB\_TEC\_HMS

```

static
js
pdfs
diabetic.pdf
src
templates
admin_dashboard.html
admin_login.html
app.html
bed_charges.html
blood_bank.html
blood_groups.html
book_appointment.html
contact.html
dashboard.html
dashboard1.html
diabetes_report.html
diabetes.html
doctors_page.html
donors.html
footer.html
heart_report.html
heart.html
index.html
login.html
medicines.html
powerbi.html
register.html
app.py

```

OUTLINE    TIMELINE

templates > blood\_bank.html

```

4   <style>
87    .advice-text h2 {
91     }
92   </style>
93   {% endblock %}
94
95   {% block content %}
96   <div class="blood-section">
97
98     <!-- Sidebar Dashboard -->
99     <div class="dashboard-sidebar">
100       
101       <div style="text-align: center; padding: 15px; color: #fff;">
102         <img alt="Dr. Umme Mezbah Akter" style="width: 170px; height: 170px; border-radius: 50%; object-fit: cover; box-shadow: 0 2px 8px rgba(0,0,0,.3); margin: 0 auto 10px auto;"/>
103         <h5 style="margin: 0; font-size: 1.1em; >Dr. Umme Mezbah Akter</h5>
104         <small style="color: #ffcd00; display: block; margin-bottom: 8px;>Specialist, Blood Bank</small>
105         <p style="margin: 0 auto; font-size: 0.9rem; max-width: 300px; line-height: 1.3em; height: 2.6em; overflow: hidden; text-overflow: ellipsis;">
106           Dr. Umme Mezbah Akter is a highly motivated specialist in Transfusion Medicine, passionate about patient care and education.
107         </p>
108       </div>
109     </div>
110     <a href="{{ url_for('blood_groups') }}>
111       <div class="dashboard-card">
112         <i class="bi bi-droplet-fill"></i>
113         <div class="stat-label" style="color: #FF6100;">Blood Groups</div>
114       </div>
115     </a>
116
117     <a href="{{ url_for('donors') }}>
118       <div class="dashboard-card">
119         <i class="bi bi-people-fill"></i>
120         <div class="stat-label" style="color: #FF6100;">Donors</div>
121     </div>
122   </div>
123 
```

## Blood\_bank

WEB\_TEC\_HMS

```

static
models
heart_model.pkl
scaler.pkl
svm_model.pkl
src
darkmode.js
dashboard.js
pdfs
diabetic.pdf
src
templates
admin_dashboard.html
admin_login.html
app.html
bed_charges.html
blood_bank.html
blood_group.html
book_appointment.html
contact.html
dashboard.html
dashboard1.html
diabetes_report.html
diabetes.html
doctors_page.html
donors.html
footer.html
heart_report.html
heart.html
index.html
login.html
medicines.html
powerbi.html

```

OUTLINE    TIMELINE

templates > blood\_group.html

```

134
135   {% block content %}
136   <div class="blood-groups-container">
137     <h2>Available Blood Groups</h2>
138     <div class="blood-navbar">
139       {% for group in groups %}
140         <div class="blood-item" data-group="{{ group.name }}>
141           {{ group.name }}
142           <span class="liters">{{ group.liters }} L</span>
143           <button class="contact-btn" onclick="openModal('{{ group.name }}')>Contact for Blood</button>
144         </div>
145       {% endfor %}
146     </div>
147   </div>
148
149   <!-- Modal -->
150   <div id="contactModal" class="modal-overlay">
151     <div class="modal">
152       <h3>Contact for Blood</h3>
153       <p>If you need blood from this group or want to donate, please contact:</p>
154       <p><strong>Phone:</strong> +1 (555) 123-4567</p>
155       <p><strong>Email:</strong> bloodbank@example.com</p>
156       <button class="close-btn" onclick="closeModal()>Close</button>
157     </div>
158   </div>
159
160   <script>
161     function openModal(bloodGroup) {
162       document.getElementById('modalBloodGroup').textContent = bloodGroup;
163       document.getElementById('contactModal').classList.add('active');
164     }
165
166     function closeModal() {
167       document.getElementById('contactModal').classList.remove('active');
168     }
169
170     // Optional: Close modal if clicked outside content
171     window.addEventListener('click', function(event) {
172       const modal = document.getElementById('contactModal');
173       if(event.target === modal) {
174         closeModal();
175       }
176     });
177   </script>

```

## Blood Groups

```

    v pdfs      96  {% endblock %}
    x diabetic.pdf  97
    v src       98  {% block content %}
    v templates 99
    o admin_dashboard.html 100 <div class="report-wrapper">
    o admin_login.html 101   <h2>Heart Disease Prediction Report</h2>
    o app.html     102
    o bed_charges.html 103   <ul>
    o blood_bank.html 104     <li><strong>Age:</strong> {{ data.age }}</li>
    o blood_groups.html 105     <li><strong>Sex:</strong> {{ data.sex }}</li>
    o book_appointment.html 106     <li><strong>Chest Pain Type:</strong> {{ data.cp }}</li>
    o contact.html 107     <li><strong>Resting BP:</strong> {{ data.trestbps }}</li>
    o dashboard.html 108     <li><strong>Cholesterol:</strong> {{ data.chol }}</li>
    o dashboard1.html 109     <li><strong>Fasting Blood Sugar:</strong> {{ data.fbs }}</li>
    o diabetes_report.html 110     <li><strong>Rest EG:</strong> {{ data.restecg }}</li>
    o diabetes.html 111     <li><strong>Max Heart Rate Achieved:</strong> {{ data.thalach }}</li>
    o doctors_page.html 112     <li><strong>Exercise Induced Angina:</strong> {{ data.exang }}</li>
    o donors.html 113     <li><strong>Oldpeak:</strong> {{ data.oldpeak }}</li>
    o footer.html 114     <li><strong>Slope:</strong> {{ data.slope }}</li>
    o heart_report.html 115     <li><strong>Ca:</strong> {{ data.ca }}</li>
    o heart.html 116     <li><strong>Thal:</strong> {{ data.thal }}</li>
    o indexhtml 117   </ul>
    o login.html 118
    o medicines.html 119   <h3>Result: {{ data.result }}</h3>
    o powerbi.html 120
    o register.html 121   <a href="/heart" class="btn-submit" style="margin-top:20px; display:inline-block;"> Predict Again</a>
    o static 122 </div>
    o stylesheets 123
    o template 124  {% endblock %}
    o static 125

```

## Heart Diseases Report

```

    v WEB_TEC_HMS
    > __pycache__
    v models
    f heart_model.pkl
    f scaler.pkl
    f svm_model.pkl
    v static
    > css
    > img
    v js
    f darkmode.js
    f dashboard.js
    v pdfs
    x diabetic.pdf
    v src
    v templates
    o admin_dashboard.html
    o admin_login.html
    o app.html
    o bed_charges.html 21  {% extends 'app.html' %} 22  {% block extra_css %} 23
    o blood_bank.html 24
    o blood_groups.html 25
    o book_appointment.html 26
    o contact.html 27
    o dashboard.html 28
    o dashboard1.html 29
    o diabetes_report.html 30
    o diabetes.html 31
    o doctors_page.html 32
    o donors.html 33
    o footer.html 34
    o heart_report.html 35
    o heart.html 36
    o indexhtml 37
    o login.html 38
    o medicines.html 39
    o powerbi.html 40
    > OUTLINE 41
    > TIMELINE 42
    0 43
    0 44

```

template > o bed\_charges.html

```

1  {% extends 'app.html' %} 2  {% block extra_css %} 3
4  <link rel="stylesheet" href="{{ url_for('static', filename='css/bed_charges.css') }}">
5
6  {% endblock %}
7  {% block content %}
8
9
10
11 <!-- bed charges.html -->
12 <div class="container my-5">
13   <h2 class="text-center mb-4" style="font-weight:700; color:white;">
14     Bed & Cabin Charges
15   </h2>
16   <p class="text-center mb-4" style="font-size: 1rem; color: white;">
17     Comprehensive list of all hospital room types, availability, daily charges, and amenities.
18 </p>
19
20 <div class="card shadow-lg p-4 border-0 rounded-4" style="background: #F9F9F9;">
21   <table class="table table-hover table-bordered align-middle text-center" style="border-radius:10px; overflow:hidden;">
22     <thead class="table-primary text-white">
23       <tr>
24         <th> Room Type</th>
25         <th> Quantity Available</th>
26         <th> Price per Day</th>
27         <th> Facilities</th>
28         <th> Notes</th>
29       </tr>
30     </thead>
31     <tbody>
32       <tr>
33         <td>General Ward</td>
34         <td>30 Beds</td>
35         <td>$ 500</td>
36         <td>Shared Room, Basic Amenities</td>
37         <td>Suitable for general patients</td>
38       </tr>
39       <tr>
40         <td>Cabin (Non-AC)</td>
41         <td>15 Rooms</td>
42         <td>$ 1,200</td>
43         <td>Private Room, TV, Fan</td>
44       </tr>

```

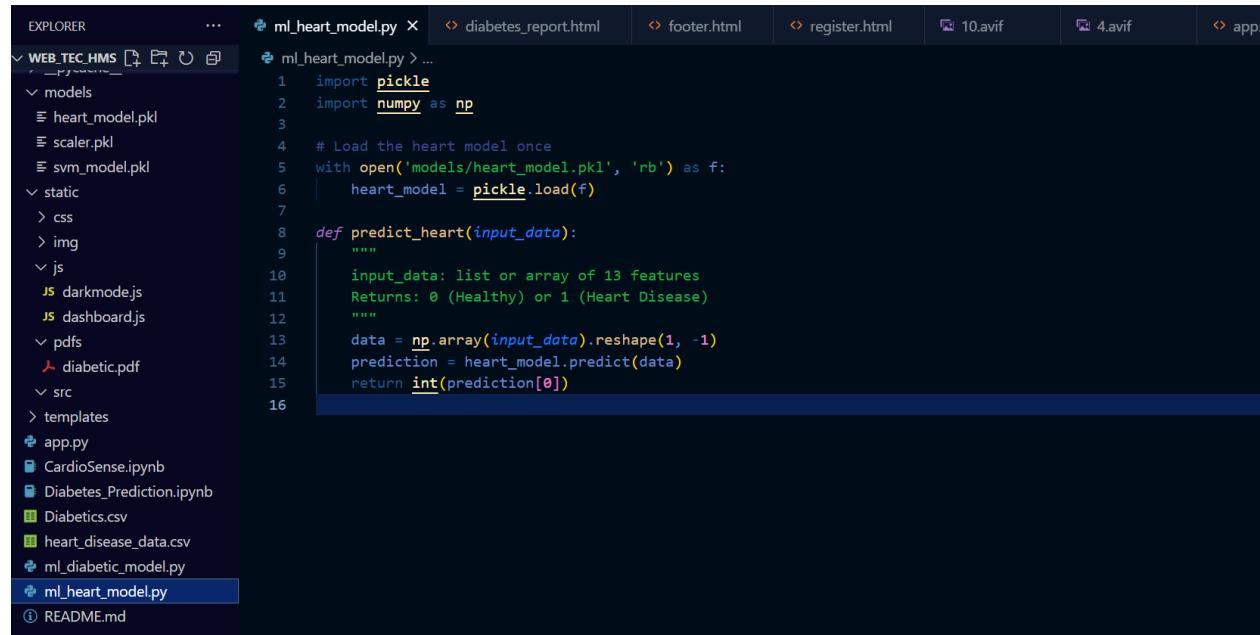
## Bed Info

```
static > JS darkmodejs > ...
1 // static/js/darkmode.js
2 document.addEventListener('DOMContentLoaded', () => {
3   const toggleBtn = document.getElementById('darkModeToggle');
4   if (!toggleBtn) return;
5
6   toggleBtn.addEventListener('click', () => {
7     document.body.classList.toggle('dark-mode');
8
9     // Toggle icon between sun/moon
10    const icon = toggleBtn.querySelector('i');
11    if (document.body.classList.contains('dark-mode')) {
12      icon.classList.replace('bi-moon-fill', 'bi-sun-fill');
13      localStorage.setItem('theme', 'dark');
14    } else {
15      icon.classList.replace('bi-sun-fill', 'bi-moon-fill');
16      localStorage.setItem('theme', 'light');
17    }
18  });
19
20 // Persist theme on reload
21 if (localStorage.getItem('theme') === 'dark') {
22   document.body.classList.add('dark-mode');
23   const icon = toggleBtn.querySelector('i');
24   if (icon) icon.classList.replace('bi-moon-fill', 'bi-sun-fill');
25 }
26 });
27
```

Darkmode.js file

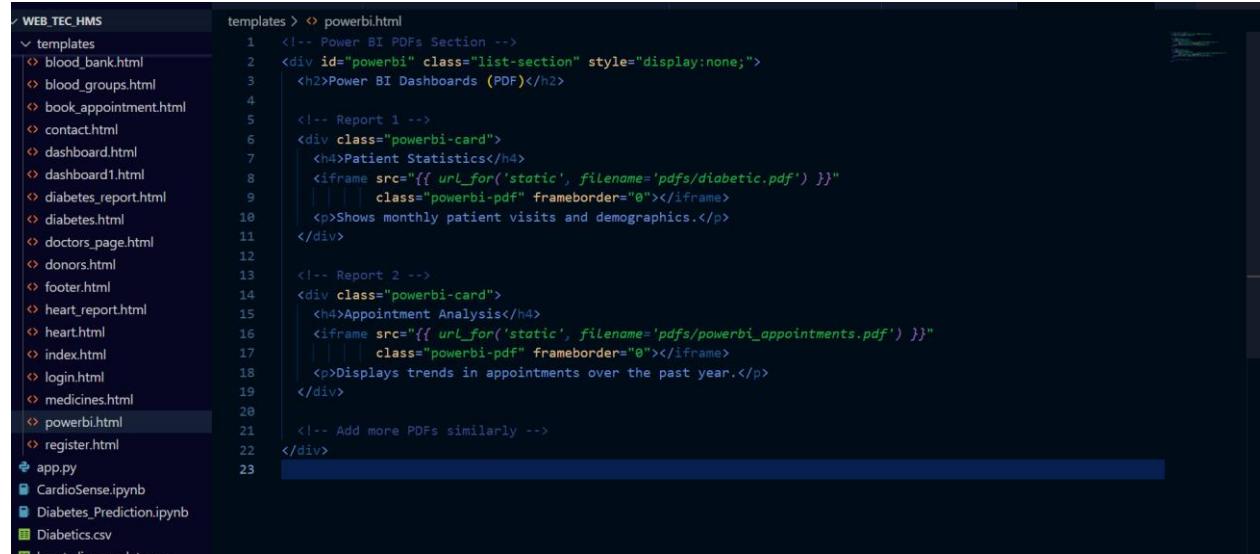
```
ml_diabetic_model.py > ...
1 import pickle
2 import numpy as np
3
4 # Load the diabetes model
5 with open('models/svm_model.pkl', 'rb') as f:
6     model = pickle.load(f)
7
8 with open('models/scaler.pkl', 'rb') as f:
9     scaler = pickle.load(f)
10
11 def predict_diabetes(input_data):
12     data = np.array(input_data).reshape(1, -1)
13     scaled_data = scaler.transform(data)
14     prediction = model.predict(scaled_data)
15     return prediction[0]
```

ML Diabetic Model.py



```
1 import pickle
2 import numpy as np
3
4 # Load the heart model once
5 with open('models/heart_model.pkl', 'rb') as f:
6     heart_model = pickle.load(f)
7
8 def predict_heart(input_data):
9     """
10         input_data: list or array of 13 features
11         Returns: 0 (Healthy) or 1 (Heart Disease)
12     """
13     data = np.array(input_data).reshape(1, -1)
14     prediction = heart_model.predict(data)
15     return int(prediction[0])
16
```

## ML Heart Model.py



```
1 <!-- Power BI PDFs Section -->
2 <div id="powerbi" class="list-section" style="display:none;">
3     <h2>Power BI Dashboards (PDF)</h2>
4
5     <!-- Report 1 -->
6     <div class="powerbi-card">
7         <h4>Patient Statistics</h4>
8         <iframe src="{{ url_for('static', filename='pdfs/diabetic.pdf') }}">
9             <!-- Power BI PDF -->
10        <p>Shows monthly patient visits and demographics.</p>
11    </div>
12
13    <!-- Report 2 -->
14    <div class="powerbi-card">
15        <h4>Appointment Analysis</h4>
16        <iframe src="{{ url_for('static', filename='pdfs/powerbi_appointments.pdf') }}">
17            <!-- Power BI PDF -->
18        <p>Displays trends in appointments over the past year.</p>
19    </div>
20
21    <!-- Add more PDFs similarly -->
22 </div>
23
```

## Power Bi file

- Prediction Model

The screenshot shows a Jupyter Notebook interface with the following details:

- Left Sidebar:** A file tree showing project structure. It includes files like `__pycache__`, `models`, `heart_model.pkl`, `scaler.pkl`, `svm_model.pkl`, `static`, `css`, `img`, `js`, `darkmode.js`, `dashboard.js`, `pdfs`, `diabetic.pdf`, `src`, `templates`, `app.py`, `CardioSense.ipynb` (selected), `Diabetes_Prediction.ipynb`, `Diabetics.csv`, `heart_disease_data.csv`, `ml_diabetic_model.py`, `ml_heart_model.py`, and `README.md`.
- Top Bar:** A toolbar with icons for file operations (New, Open, Save, etc.) and notebook controls (Generate, Code, Markdown, Run All, Clear All Outputs, Outline).
- Title Bar:** The title is "CardioSense.ipynb > Predicting system".
- Content Area:** The main area displays a Python script for a prediction system.

```
# input random data for testing
input_data = [62.0, 0, 140, 28.0, 0, 160.0, 3.6, 0, 2, 2] # Converted to a list
# Get the feature names from the training data
feature_names = X_train.columns
input_df = pd.DataFrame([input_data], columns=feature_names) # Create a DataFrame with feature names

prediction = model.predict(input_df) # Use the DataFrame for prediction
print(prediction)

if(prediction[0]==0):
    print('The person is Healthy')
else:
    print('The Person is Sick')

...
[0]
The person is Healthy

import pickle
from google.colab import files

# Assuming your trained model is named 'model'
with open('heart_model.pkl', 'wb') as file:
    pickle.dump(model, file)

print("heart_model.pkl saved ✅")

# Download the model
files.download('heart_model.pkl')
```
- Bottom Right:** A small "Python" icon.

## Heart Diseases

The screenshot shows a Jupyter Notebook interface with several open files:

- EXPLORER**: Shows the project structure with files like `Diabetes_Prediction.ipynb`, `Diabetics.csv`, `heart_disease_data.csv`, `CardioSense.ipynb`, `heart_model.pkl`, `dashboard.css`, `heart.css`, and `sick.wl`.
- WEB\_TEC\_HMS**: A sidebar showing file operations like `> _pycache_`, `models`, `static`, `css`, `img`, `js` (with `darkmode.js` and `dashboard.js`), `pdfs` (with `diabetic.pdf`), `src`, `templates`, `app.py`, and `CardioSense.ipynb`.
- Diabetes\_Prediction.ipynb**: The active notebook tab. It contains Python code for loading data, standardizing it, making predictions, and printing results. It also includes code for saving the trained SVM model and its scaler to files.
- Diabetics.csv**, **heart\_disease\_data.csv**, **ml\_diabetic\_model.py**, **ml\_heart\_model.py**, and **README.md**: These are listed in the Explorer sidebar.

The code in the `Diabetes_Prediction.ipynb` cell is as follows:

```
input_data = (1 ,85, 66, 29, 0 ,26.6, 0.351, 31 )
# convert in np array
check_data = np.asarray(input_data)
#reshape
check_data_reshape=check_data.reshape(1,-1)
# standardize the input data
std_data=loaded_scaler.transform(check_data_reshape)
print(std_data)
prediction=loaded_model.predict(std_data)
print(prediction)
if(prediction[0]==0):
    print("The person is not diabetic")
else:
    print("The person is diabetic")
```

The code in the second cell is as follows:

```
from google.colab import files

# 7. Save model to 'svm_model.pkl'
with open('svm_model.pkl', 'wb') as f:
    pickle.dump(classifier, f)

# 8. Save scaler to 'scaler.pkl'
with open('scaler.pkl', 'wb') as f:
    pickle.dump(scalar, f)

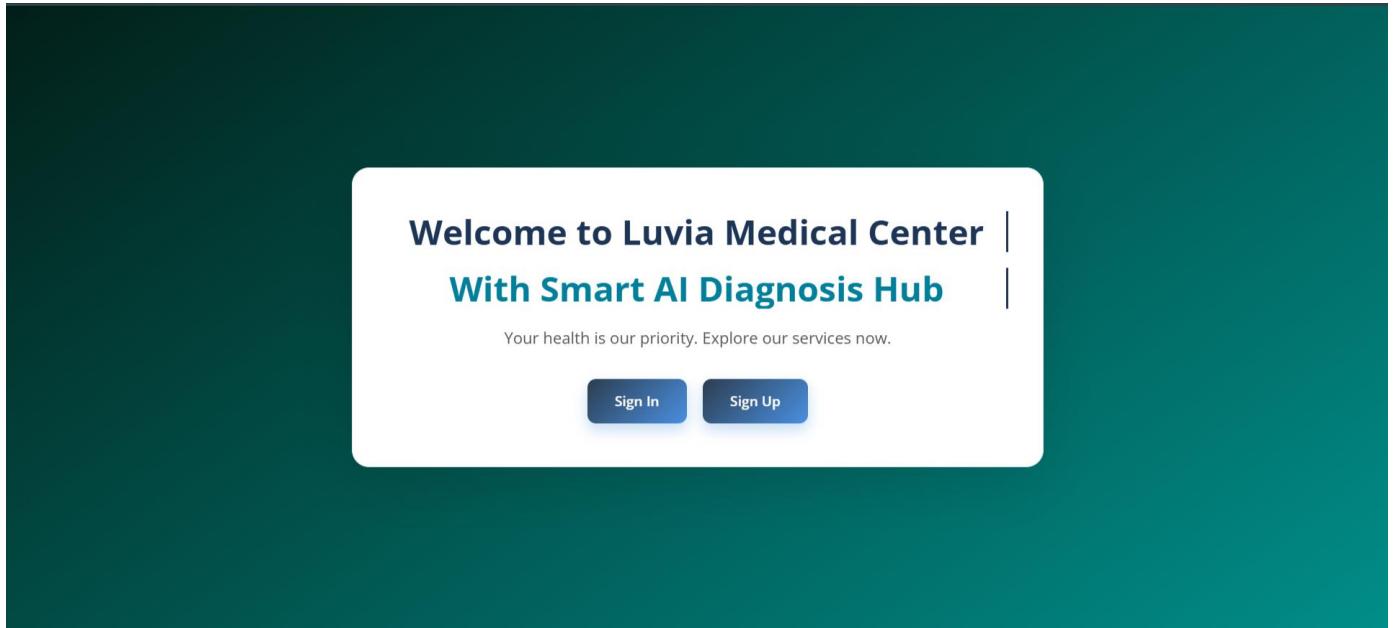
# 9. Download both files
files.download('svm_model.pkl')
files.download('scaler.pkl')
```

Diabetic

## 4. Output

---

- User Interface



- User Register Form

The left side shows a "Register form" interface with three input fields: "Name", "Email", and "Password", each with a corresponding text input box. Below the inputs are two buttons: a dark blue "Register" button and a light gray "Login page" button. To the right is a photograph of a medical team of doctors and nurses attending to a patient in an intensive care unit (ICU) setting.

- User Login Form



**Login**

Email

Password

⌚ Opening Time: 9:00 AM | ⚓ Closing Time: 6:00 PM

 Luvia Medical Center Appointments Doctors

**Welcome to the Dashboard**

**User Information**

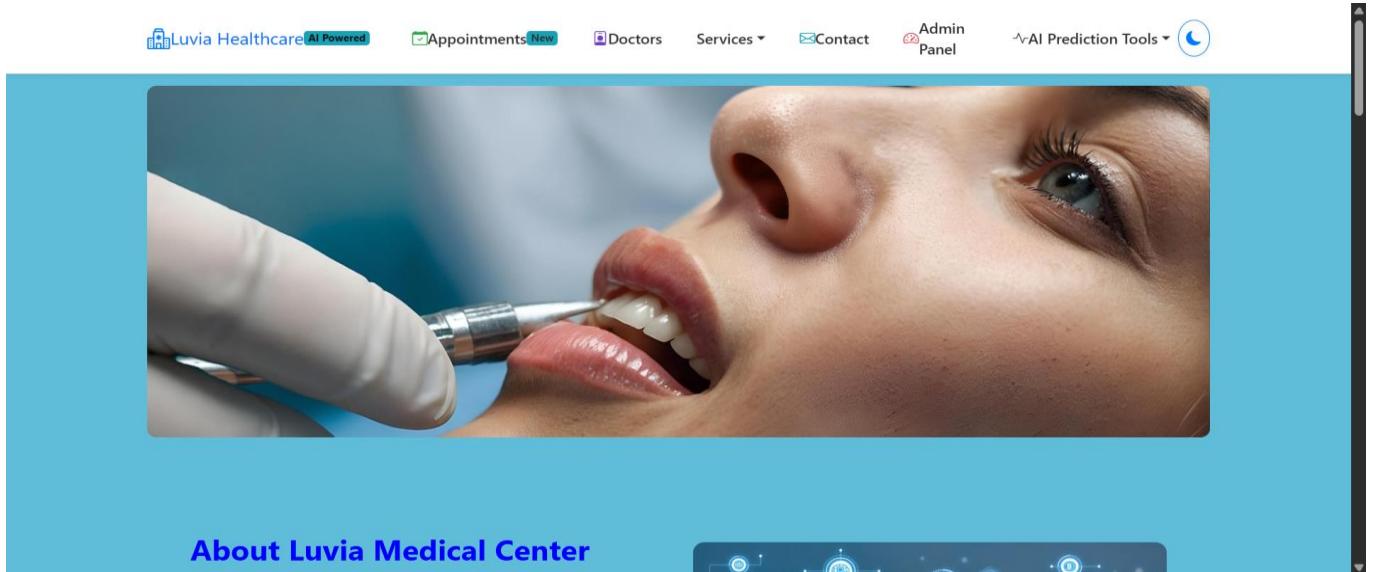
Name: reja

Email: reja@gmail.com

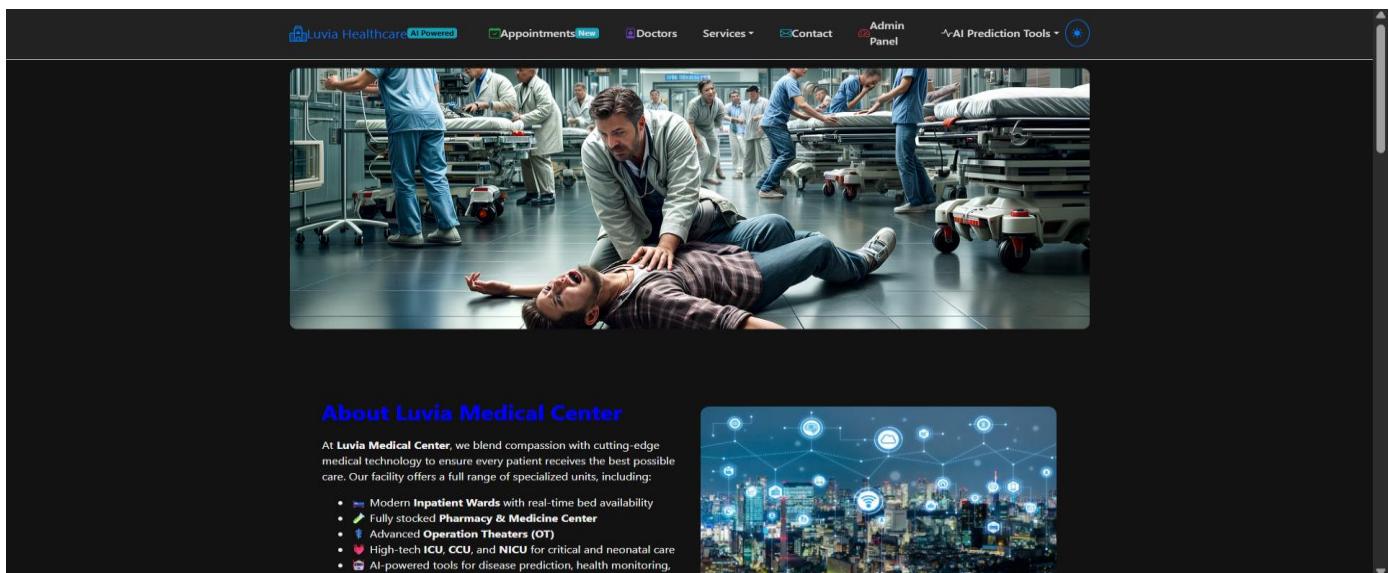
[Logout](#) [Visit Dashboard](#)

127.0.0.1:5000/dashboard

- Dashboard



The screenshot shows a medical dashboard for Luvia Healthcare. At the top, there's a navigation bar with links for Appointments (New), Doctors, Services, Contact, Admin Panel, and AI Prediction Tools. Below the navigation is a large image of a dental procedure where a patient's teeth are being treated. A blue banner at the bottom of the screen reads "About Luvia Medical Center".



The screenshot shows a medical dashboard for Luvia Healthcare. At the top, there's a navigation bar with links for Appointments (New), Doctors, Services, Contact, Admin Panel, and AI Prediction Tools. Below the navigation is a large image of a medical emergency in an operating room where a patient is lying on the floor and a doctor is attending to them. A blue banner at the bottom of the screen reads "About Luvia Medical Center". To the right of the banner is a graphic of a city skyline with various icons representing medical and technological integration.

## About Luvia Medical Center

At Luvia Medical Center, we blend compassion with cutting-edge medical technology to ensure every patient receives the best possible care. Our facility offers a full range of specialized units, including:

- ➡ Modern Inpatient Wards with real-time bed availability
- ➡ Fully stocked Pharmacy & Medicine Center
- ➡ Advanced Operation Theaters (OT)
- ➡ High-tech ICU, CCU, and NICU for critical and neonatal care
- ➡ AI-powered tools for disease prediction, health monitoring, and diagnostics



## Our Facilities

We constantly invest in high-end equipment



- ➡ 24/7 Emergency & ICU
- ➡ Modern Diagnostic Lab
- ➡ Experienced Doctors & Nurses
- ➡ Free Health Checkup Camps
- ➡ Online Appointment & Report

## 💡 AI-Powered Disease Prediction

Use our advanced machine learning models to assess your health risk and get preventive guidance instantly.

### Heart Disease

Check your heart risk with just a few inputs.

[Predict Now](#)

### Diabetes

Get personalized diabetes risk assessment.

[Predict Now](#)

### Cancer

Early detection predictions using ML models.

[Predict Now](#)

### COVID-19

Predict your COVID-19 risk based on symptoms.

[Predict Now](#)

## ⚠️ Emergency Care

Our emergency department is open 24/7 to provide critical care and life-saving support.

### 24/7 ER

Fully staffed emergency room ready for any critical situation.

### Trauma & Critical Care

Advanced trauma and critical care units for immediate treatment.

### Ambulance Service

Rapid response ambulance & mobile ICU for emergencies.

[📞 Call Emergency Now](#)

- Book an Appointment

- Blood Bank

**Dr. Umme Mezbah Akter**  
MBBS, MD (Transfusion Medicine)  
*Specialist, Blood Bank*

Dr. Umme Mezbah Akter is a highly motivated specialist in Transfusion

Blood Groups

Donors

"One donation can save up to three lives — your blood is precious, share it."

**Blood Donation Saves Lives**

Donating blood is a simple act of kindness that can make a big difference. Your donation helps patients in emergencies, surgeries, and chronic illnesses. Be a hero — donate blood regularly and help save lives in your community.



## 滴 Donor List

[+ Add Donor](#)

<b>Farhana Anie (B+)</b> Email: mimi12@gmail.com Phone: 01811685234 Last Donation: 2025-07-28	<b>Richmond Antor Biswas (B+)</b> Email: rich@gmail.com Phone: 01811685119 Last Donation: 2025-06-02	<b>Istiak Mamun (A-)</b> Email: mamun@gmail.com Phone: 01811685234 Last Donation: 2024-01-30
<b>Tasnim Tushi (B+)</b> Email: tushi@gmail.com Phone: 01811685143 Last Donation: 2025-04-27	<b>Nikolas Khan (AB-)</b> Email: khan@gmail.com Phone: 01811685119 Last Donation: 2025-06-30	

## Donor List

[+ Add Donor](#)

**Donor Name**

**Email**

**Phone**

**Blood Group**

**Last Donation Date**  
 Select

**Health Condition**

**Submit**

**Farhana Anie (B+)**  
 Email: mimi12@gmail.com  
 Phone: 01811685234  
 Last Donation: 2025-07-28

**Richmond Antor Biswas (B+)**  
 Email: rich@gmail.com  
 Phone: 01811685119  
 Last Donation: 2025-06-02

**Istiak Mamun (A-)**  
 Email: mamun@gmail.com  
 Phone: 01811685234  
 Last Donation: 2024-01-30

**Tasnim Tushi (B+)**  
 Email: tasnimtushi@gmail.com  
 Phone: 01811685234  
 Last Donation: 2025-07-28

**Nikolas Khan (AB-)**  
 Email: nikolas.khan@gmail.com  
 Phone: 01811685234  
 Last Donation: 2024-01-30

- Bed Information

### 🛏️ Bed & Cabin Charges

Comprehensive list of all hospital room types, availability, daily charges, and amenities.

🛏️ Room Type	🛌 Quantity Available	₹ Price per Day	⭐ Facilities	📝 Notes
General Ward	30 Beds	₹ 500	Shared Room, Basic Amenities	Suitable for general patients
Cabin (Non-AC)	15 Rooms	₹ 1,200	Private Room, TV, Fan	Good for semi-private stay
Cabin (AC)	20 Rooms	₹ 2,000	Private Room, AC, TV, Attached Bathroom	Comfortable for long-term stay
ICU	10 Beds	₹ 5,000	Advanced Monitoring, Oxygen, Ventilator	Critical care patients only
CCU	8 Beds	₹ 6,000	Cardiac Monitoring, Specialist Care	For cardiac patients
VIP Suite	5 Rooms	₹ 10,000	Private Suite, AC, TV, Mini Fridge, Attached Bathroom	Luxury stay for VIP patients

⚠️ All charges are per day. Taxes and additional services may apply. For reservations and inquiries, please contact hospital administration.

- Contact Us

## Contact Us

We'd love to hear from you

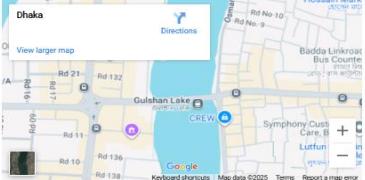


Your Name

Your Email

Your Message

**Send Message**



**Reach Us At**  
Email: support@example.com  
Phone: +880-123-456-7890  
Address: 123 Gulshan Avenue, Dhaka, Bangladesh

- Admin Login Form

## Admin Login

**Email Address**

**Password**

**Login**

- Admin Dashboard

**Welcome, janie@example.com 🙌**

### Admin List

ID	Username	Email	Created At
1	janie	janie@example.com	2025-08-12 01:57:53

**Admin Menu**

- Admin List**
- Doctors
- Appointment List
- User List
- Donor List
- Contact List
- Power BI
- Logout

**Welcome, janie@example.com 🙌**

### Doctors List

Add Doctor					
ID	Name	Email	Phone	Specialty	Photo
1	Mr.hasan	hasan@gmail.com	01811685143	Medicine	
2	Mridul Datta	datta@gmail.com	01834567234	ENT	

**Admin Menu**

- Doctors**
- Appointment List
- User List
- Donor List
- Contact List
- Power BI
- Logout

Welcome, janie@example.com 🙌

### Doctors List

Add Doctor

Name:

Email:

Phone:

Specialty:

Photo:  No file chosen

Add Doctor

ID	Name	Email	Phone	Specialty	Photo
1	Mr.hasan	hasan@gmail.com	01811685143	Medicine	

Welcome, janie@example.com 🙌

### Appointment List

ID	Name	Email	Phone	Date	Time	Doctor
1	Faria Jahan Janie	janiefaria315@gmail.com	01811685119	2025-08-07	15:04:00	Dr. Ahmed
2	reja	reja@gmail.com	01811685119	2025-07-29	4:22:00	Dr. Rahman
3	jannat hasan	hasan@gmail.com	01834567234	2025-08-30	19:36:00	Dr. Rahman

**Welcome, janie@example.com 🙌**

### User List

ID	Name	Email
3	Antor	antor@gmail.com
15	janie	janie@gmail.com
16	fdh	janiefaria31hghhhhh5@gmail.com
17	gias	gias4@gmail.com
18	er	a@gmail.com
19	israt	israt@gmail.com
21	reja	reja@gmail.com

**Welcome, janie@example.com 🙌**

### Donor List

ID	Name	Email	Phone	Blood Group
1	Nikolas Khan	khan@gmail.com	01811685119	AB-
2	Tasnim Tushi	tushi@gmail.com	01811685143	B+
3	Istiak Mamun	mamun@gmail.com	01811685234	A-
4	Richmond Antor Biswas	rich@gmail.com	01811685119	B+
5	Farhana Anie	mimi12@gmail.com	01811685234	B+

**Welcome, janie@example.com 🙌**

### Contact List

ID	Name	Email	Message	Submitted At
1	reja	reja@gmail.com	hfqj	2025-08-04 22:33:30
2	mimi	mimi12@gmail.com	I need info about bloodbank	2025-08-13 22:41:45

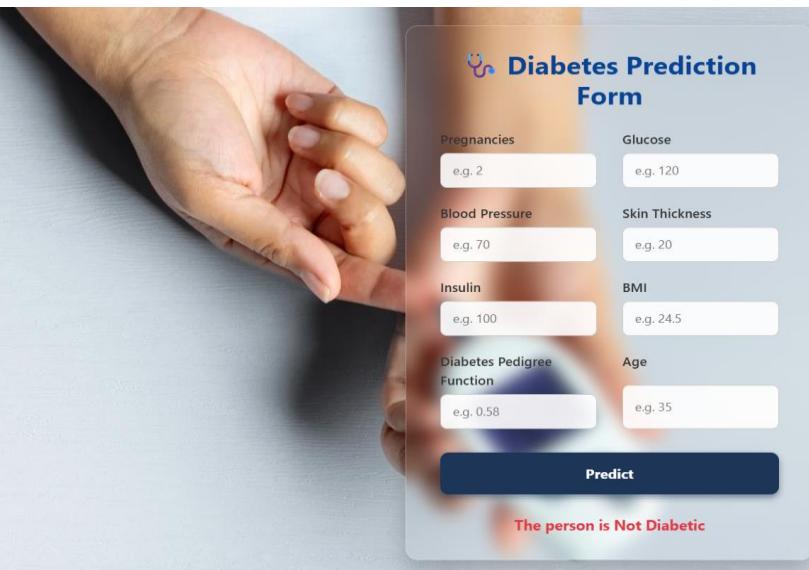
- Predict Diabetic:



**Diabetes Prediction Form**

Pregnancies e.g. 2	Glucose e.g. 120
Blood Pressure e.g. 70	Skin Thickness e.g. 20
Insulin e.g. 100	BMI e.g. 24.5
Diabetes Pedigree Function e.g. 0.58	Age e.g. 35

**Predict**



**Show Report**

**Diabetes Prediction Form**

Pregnancies e.g. 2	Glucose e.g. 120
Blood Pressure e.g. 70	Skin Thickness e.g. 20
Insulin e.g. 100	BMI e.g. 24.5
Diabetes Pedigree Function e.g. 0.58	Age e.g. 35

**Predict**

The person is Not Diabetic



**Prediction Report**

Pregnancies:	2
Glucose:	140
Blood Pressure:	80
Skin Thickness:	30
Insulin:	100
BMI:	25.9
Diabetes Pedigree Function:	0.55
Age:	40

**Result:** The person is Not Diabetic

- Predict Heart Diseases:



### Heart Disease Prediction Form

Age e.g. 55	Sex (1=Male, 0=Female) e.g. 1	Chest Pain Type (0-3) e.g. 2
Resting BP e.g. 130	Cholesterol e.g. 250	Fasting Blood Sugar >120 mg/dl (1=yes, 0=no) e.g. 0
Rest ECG (0-2) e.g. 1	Max Heart Rate Achieved e.g. 150	Exercise Induced Angina (1=yes, 0=no) e.g. 0
Oldpeak e.g. 1.5	Slope (0-2) e.g. 1	Ca (0-4) e.g. 0
Thal (1=Normal,2=Fixed defect,3=Reversible defect) e.g. 2		
<b>Predict</b>		

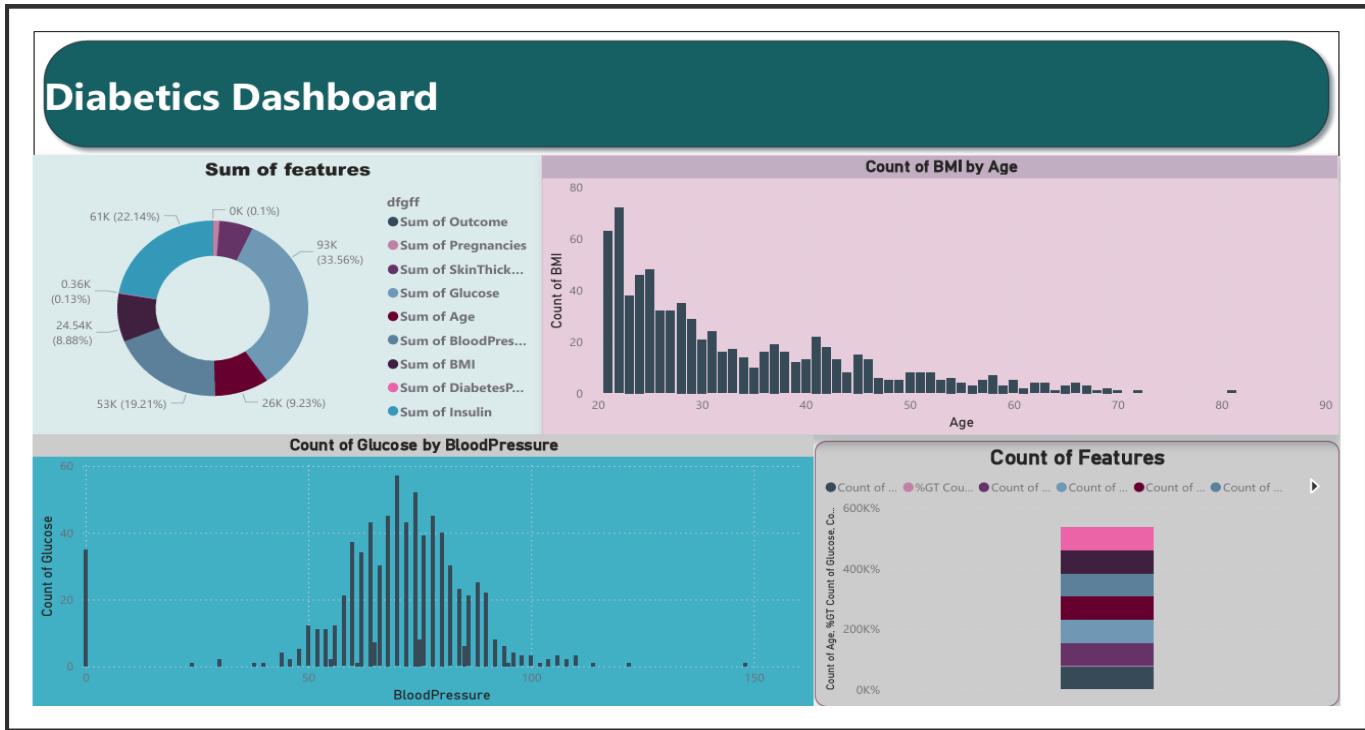
### Heart Disease Prediction Report

Age:	60
Sex:	1
Chest Pain Type:	2
Resting BP:	150
Cholesterol:	300
Fasting Blood Sugar:	1
Rest ECG:	2
Max Heart Rate Achieved:	120
Exercise Induced Angina:	0
Oldpeak:	1.9
Slope:	2
Ca:	4
Thal:	3

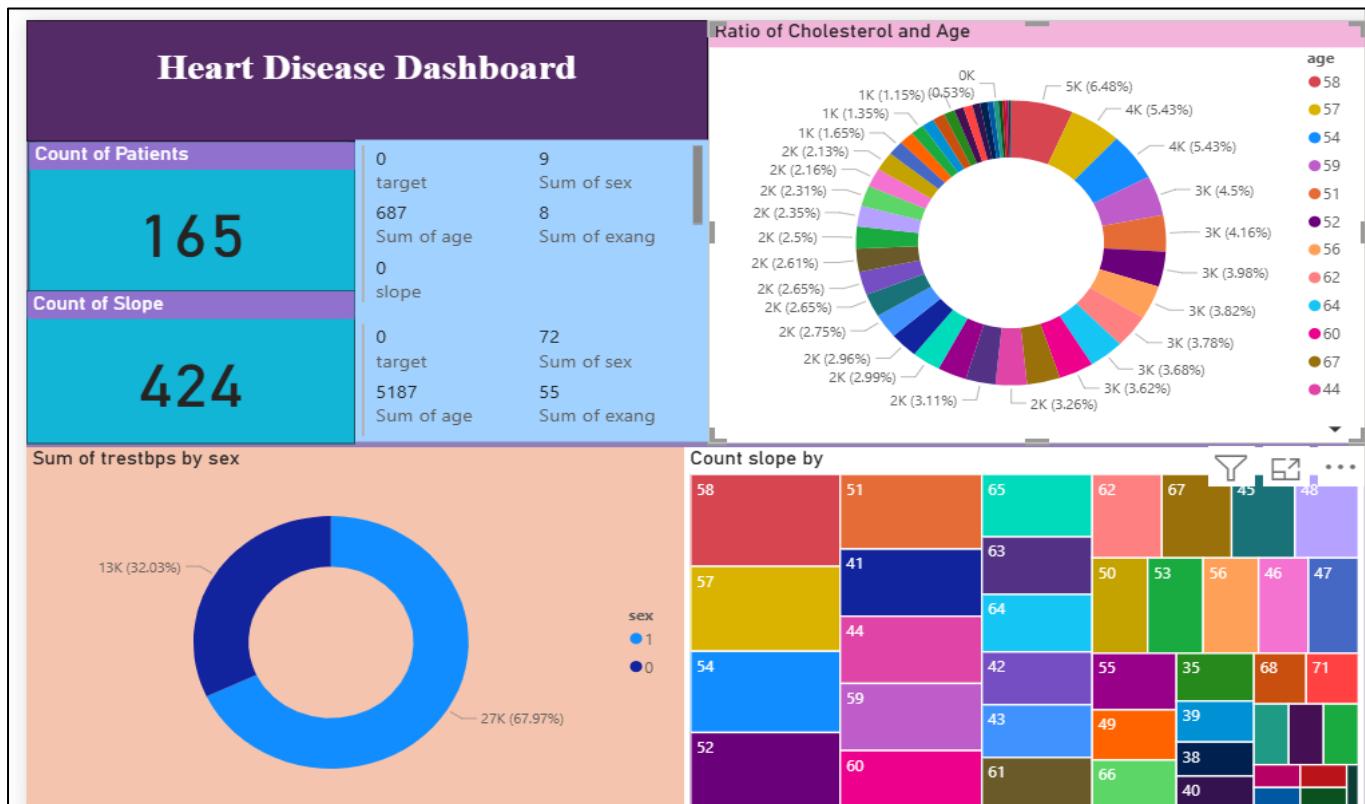
**Result: The person is Healthy**

**Predict Again**

- PowerBI :



Diabetic Dashboard



Heart Disease Dashboard

## **5. Conclusion**

---

This project successfully demonstrates the use of Flask to build a complete hospital management and prediction system. It integrates different technologies such as Python, Flask, MySQL, and machine learning to provide reliable healthcare-related services in one platform. The system includes features like patient management, doctor information, health tips, bed availability, and prediction models for diseases such as diabetes, cancer, and pregnancy. By combining data visualization and machine learning, the project helps in offering quick and effective decision support for both patients and hospital staff. The use of Flask makes the application lightweight, fast, and easy to maintain, while the database ensures secure and organized storage of information. This hospital system not only provides medical predictions but also focuses on improving patient care by offering emergency care options, doctor lists, and contact facilities. Overall, the project highlights how technology can be applied in the healthcare sector to make hospital services smarter, more accessible, and more efficient. It also creates scope for future improvements like adding more advanced prediction models, real-time data updates, and expanding the system into a fully digital healthcare solution.