

Hospital Management System

22.12.2022

—

Certificate

Dec 22, 2022

We are pleased to present this Certificate of Achievement to Tanuku Chidroop, in recognition of their outstanding work on the project report on "Hospital Management System."

They have demonstrated exceptional dedication, hard work, and technical expertise in the completion of this project. Their research, analysis, and communication skills were of the highest caliber, and the final project report was a testament to their commitment to excellence.

We are proud to acknowledge their achievement and look forward to their continued success in the future.

Supervisor,

Ms. Inturi Anitha Rani

Assistant Professor

SRM University - AP



Acknowledgement

We would like to express our sincere gratitude to [Name of Instructor or Institution] for providing us with the opportunity to work on this project. Their guidance and support have been invaluable in helping us to complete this report.

We would also like to thank our seniors for their assistance and support. Without their help, this project would not have been possible.

Finally, we would like to thank our friends and family for their encouragement and support throughout the duration of this project.

Sincerely,

Tanuku Chidroop - AP20110010154

Abstract

Hospital Management System is an organized computerized system designed and programmed to deal with day to day operations and management of the hospital activities. The program can look after inpatients, outpatients, records, database treatments, status illness, billings in the pharmacy and labs. It also maintains hospital information such as ward id, doctors in charge and department administering. The major problem for the patient nowadays is to get a report after consultation. Many hospitals manage reports in their system but it's not available to the patient when he / she is outside. In this project we are going to provide the extra facility to store the report in the database and make it available from anywhere in the world.

Table of Contents

Certificate	1
Acknowledgement	2
Abstract	3
Table of Contents	4
Introduction	5
ER Diagram	6
Code	7
Site Snapshots	16



Introduction

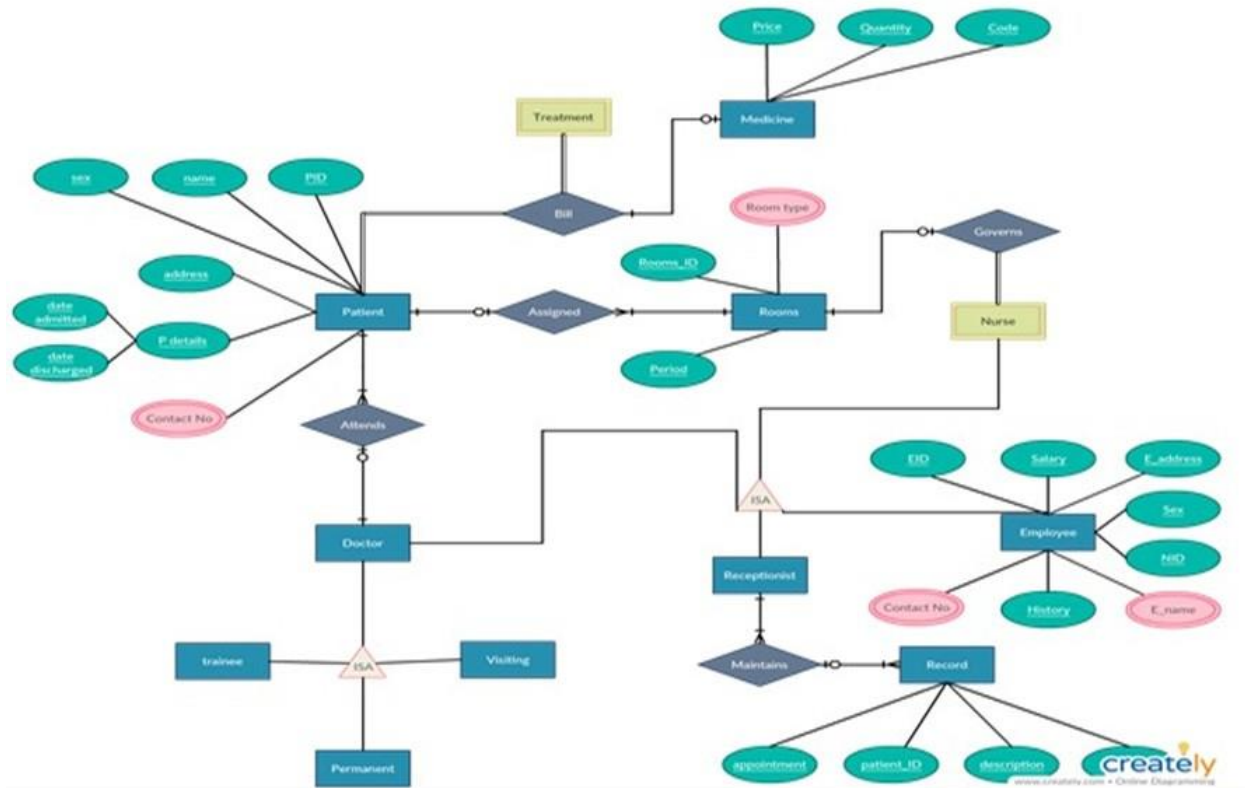
The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

ER Diagram



Code

The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders like static, templates, v2, and venv, and files like 1.py through 11.py. The main.py file is selected. The code editor shows the following code:

```

1  from flask import Flask,render_template,request,session,redirect,url_for,flash
2  from flask_sqlalchemy import SQLAlchemy
3  from flask_login import UserMixin
4  from werkzeug.security import generate_password_hash,check_password_hash
5  from flask_login import login_user,logout_user,login_manager,LoginManager
6  from flask_login import login_required,current_user
7  from flask_mail import Mail
8  import json
9
10
11
12  # MY db connection
13  local_server= True
14  app = Flask(__name__)
15  app.secret_key='aneeqah'
16
17
18  # this is for getting unique user access
19  login_manager=LoginManager(app)
20  login_manager.login_view='login'
21
22  # SMTP MAIL SERVER SETTINGS
23
24  app.config.update(
25      MAIL_SERVER='smtp.gmail.com',
26      MAIL_PORT='465',
27      MAIL_USE_SSL=True,
28      MAIL_USERNAME="add your_email-id".

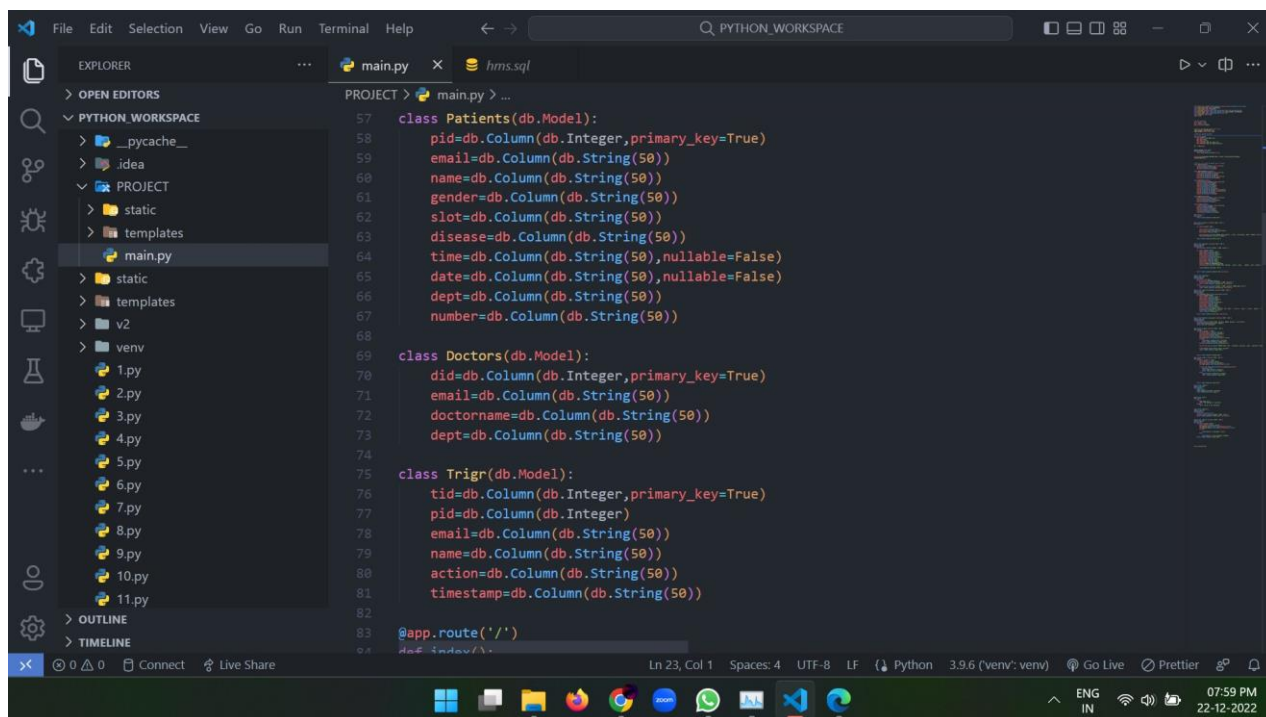
```

The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders like static, templates, v2, and venv, and files like 1.py through 11.py. The main.py file is selected. The code editor shows the following code:

```

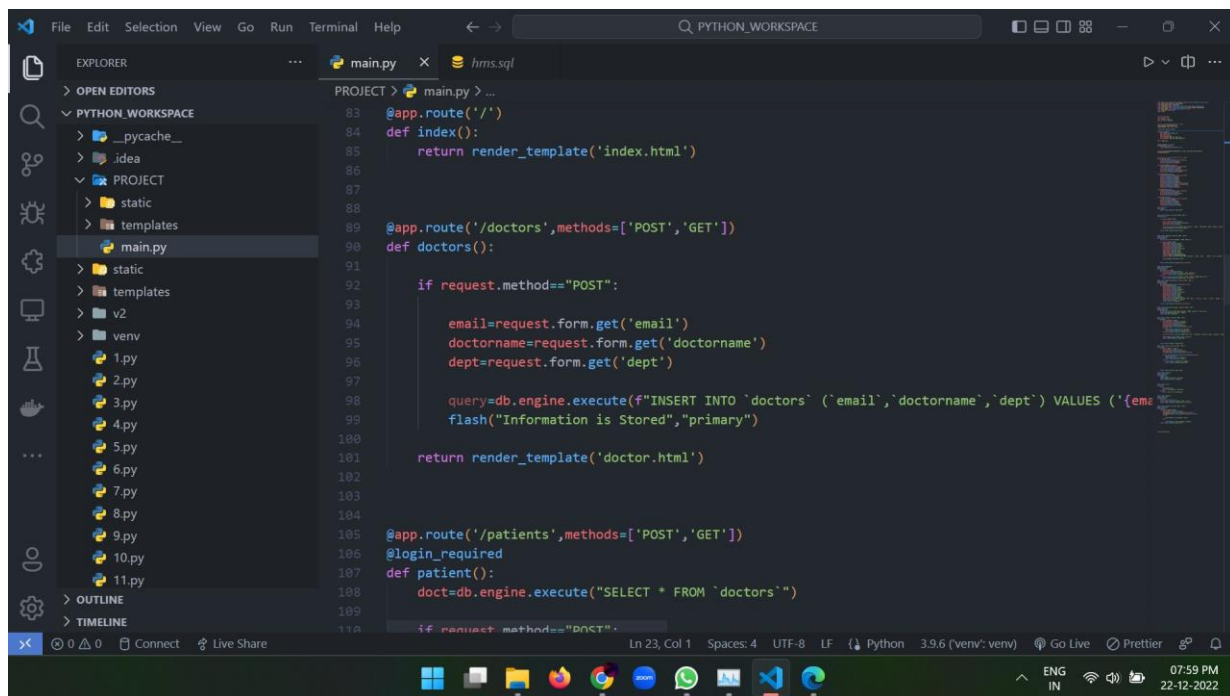
28      MAIL_USERNAME="add your gmail-id",
29      MAIL_PASSWORD="add your gmail-password"
30  )
31  mail = Mail(app)
32
33
34  @login_manager.user_loader
35  def load_user(user_id):
36      return User.query.get(int(user_id))
37
38
39  app.config['SQLALCHEMY_DATABASE_URI']='mysql://root:@localhost/hmdbms'
40  db=SQLAlchemy(app)
41
42
43  # here we will create db models that is tables
44  class Test(db.Model):
45      id=db.Column(db.Integer,primary_key=True)
46      name=db.Column(db.String(100))
47      email=db.Column(db.String(100))
48
49
50  class User(UserMixin,db.Model):
51      id=db.Column(db.Integer,primary_key=True)
52      username=db.Column(db.String(50))
53      usertype=db.Column(db.String(50))
54      email=db.Column(db.String(50),unique=True)
55      password=db.Column(db.String(100))

```

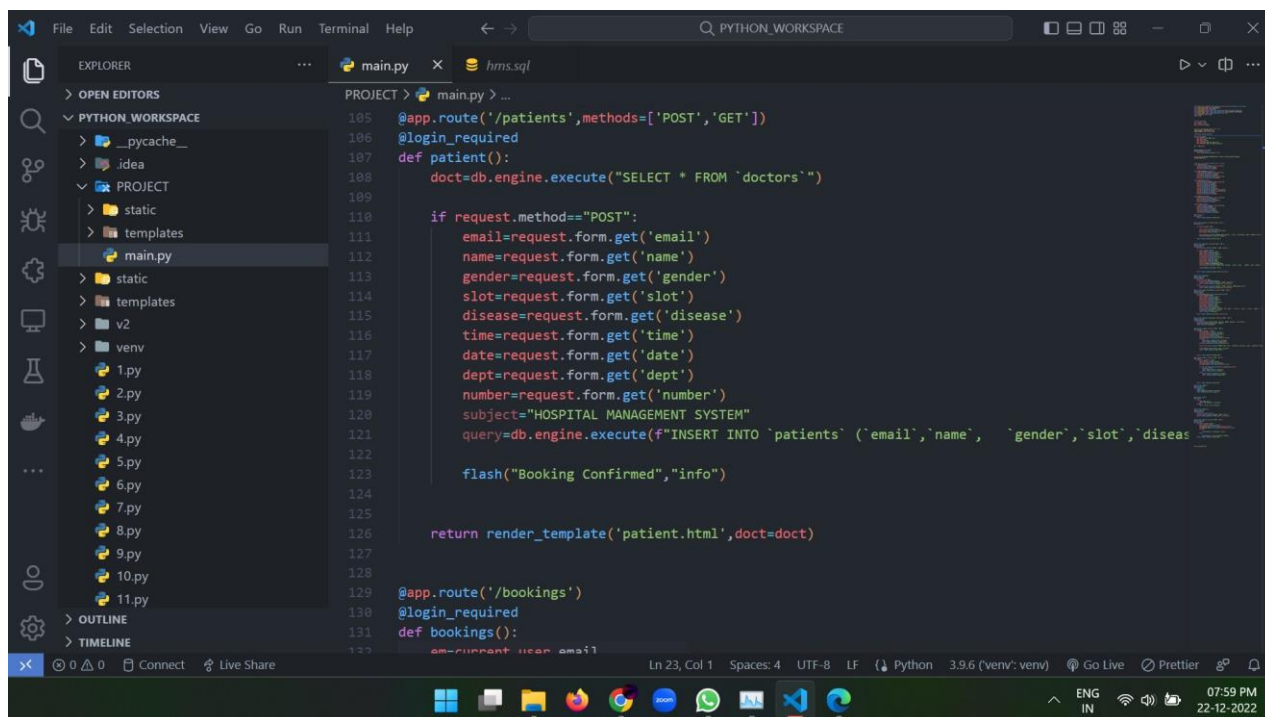
The screenshot shows the VS Code editor with the `main.py` file open. The Explorer panel on the left shows the project structure with folders like `static`, `templates`, and `venv`. The main editor displays the following code:

```
57 class Patients(db.Model):
58     pid=db.Column(db.Integer,primary_key=True)
59     email=db.Column(db.String(50))
60     name=db.Column(db.String(50))
61     gender=db.Column(db.String(50))
62     slot=db.Column(db.String(50))
63     disease=db.Column(db.String(50))
64     time=db.Column(db.String(50),nullable=False)
65     date=db.Column(db.String(50),nullable=False)
66     dept=db.Column(db.String(50))
67     number=db.Column(db.String(50))
68
69 class Doctors(db.Model):
70     did=db.Column(db.Integer,primary_key=True)
71     email=db.Column(db.String(50))
72     doctorname=db.Column(db.String(50))
73     dept=db.Column(db.String(50))
74
75 class Trigr(db.Model):
76     tid=db.Column(db.Integer,primary_key=True)
77     pid=db.Column(db.Integer)
78     email=db.Column(db.String(50))
79     name=db.Column(db.String(50))
80     action=db.Column(db.String(50))
81     timestamp=db.Column(db.String(50))
82
83 @app.route('/')
84 def index():
85     return render_template('index.html')
```

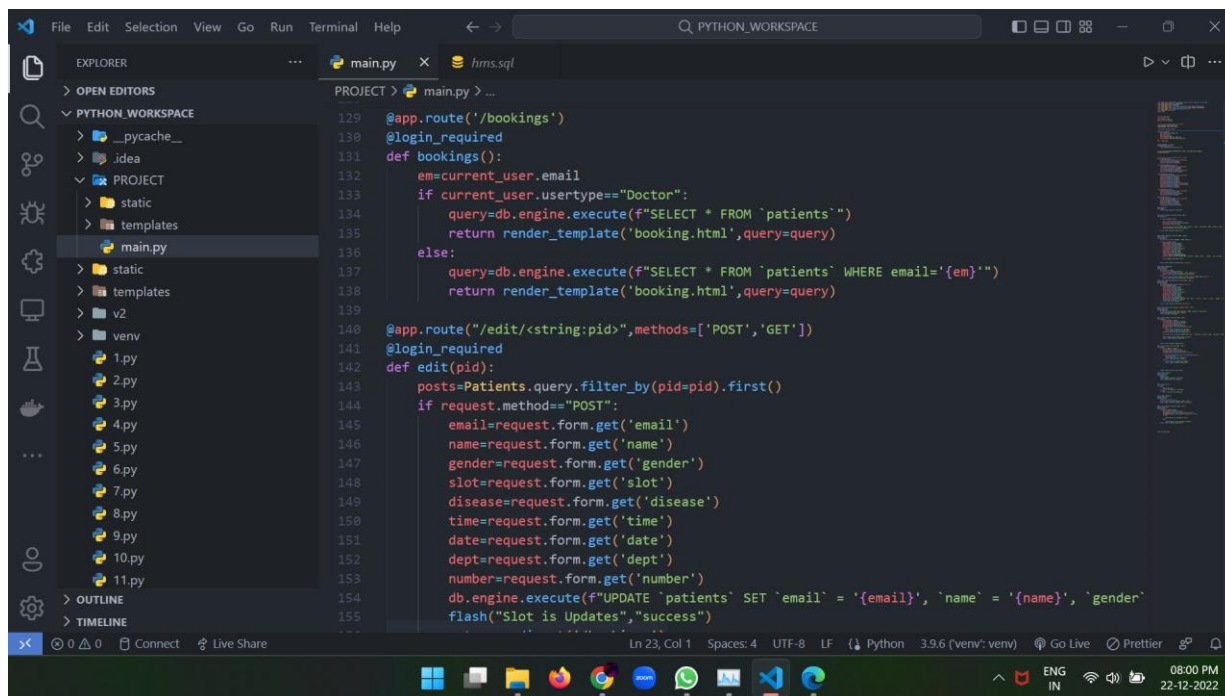


The screenshot shows the VS Code editor with the `main.py` file open. The Explorer panel on the left shows the project structure. The main editor displays the following code:

```
83 @app.route('/')
84 def index():
85     return render_template('index.html')
86
87
88
89 @app.route('/doctors',methods=['POST','GET'])
90 def doctors():
91
92     if request.method=="POST":
93
94         email=request.form.get('email')
95         doctorname=request.form.get('doctorname')
96         dept=request.form.get('dept')
97
98         query=db.engine.execute(f"INSERT INTO `doctors` (`email`,`doctorname`,`dept`) VALUES ('{email}','{doctorname}','{dept}')")
99         flash("Information is Stored","primary")
100
101     return render_template('doctor.html')
102
103
104
105 @app.route('/patients',methods=['POST','GET'])
106 @login_required
107 def patient():
108     doct=db.engine.execute("SELECT * FROM `doctors`")
109
110     if request.method=="POST":
```



```
105 @app.route('/patients', methods=['POST', 'GET'])
106 @login_required
107 def patient():
108     doct=db.engine.execute("SELECT * FROM `doctors`")
109
110     if request.method=="POST":
111         email=request.form.get('email')
112         name=request.form.get('name')
113         gender=request.form.get('gender')
114         slot=request.form.get('slot')
115         disease=request.form.get('disease')
116         time=request.form.get('time')
117         date=request.form.get('date')
118         dept=request.form.get('dept')
119         number=request.form.get('number')
120         subject="HOSPITAL MANAGEMENT SYSTEM"
121         query=db.engine.execute(f"INSERT INTO `patients` (`email`,`name`,`gender`,`slot`,`disease`,`time`,`date`,`dept`,`number`,`subject`) VALUES ('{email}','{name}','{gender}','{slot}','{disease}','{time}','{date}','{dept}','{number}','{subject}')")
122
123         flash("Booking Confirmed","info")
124
125     return render_template('patient.html',doct=doct)
126
127
128
129 @app.route('/bookings')
130 @login_required
131 def bookings():
132     em=current_user.email
```



```
129 @app.route('/bookings')
130 @login_required
131 def bookings():
132     em=current_user.email
133     if current_user.usertype=="Doctor":
134         query=db.engine.execute(f"SELECT * FROM `patients`")
135         return render_template('booking.html',query=query)
136     else:
137         query=db.engine.execute(f"SELECT * FROM `patients` WHERE email='{em}'")
138         return render_template('booking.html',query=query)
139
140 @app.route("/edit/<string:pid>", methods=['POST', 'GET'])
141 @login_required
142 def edit(pid):
143     posts=Patients.query.filter_by(pid=pid).first()
144     if request.method=="POST":
145         email=request.form.get('email')
146         name=request.form.get('name')
147         gender=request.form.get('gender')
148         slot=request.form.get('slot')
149         disease=request.form.get('disease')
150         time=request.form.get('time')
151         date=request.form.get('date')
152         dept=request.form.get('dept')
153         number=request.form.get('number')
154         db.engine.execute(f"UPDATE `patients` SET `email` = '{email}', `name` = '{name}', `gender` = '{gender}', `slot` = '{slot}', `disease` = '{disease}', `time` = '{time}', `date` = '{date}', `dept` = '{dept}', `number` = '{number}' WHERE pid={pid}")
155         flash("Slot is Updates","success")
```

The screenshot shows the VS Code editor with the `main.py` file open. The Explorer sidebar on the left shows the project structure with folders like `static`, `templates`, and `venv`. The main editor area displays the following Python code:

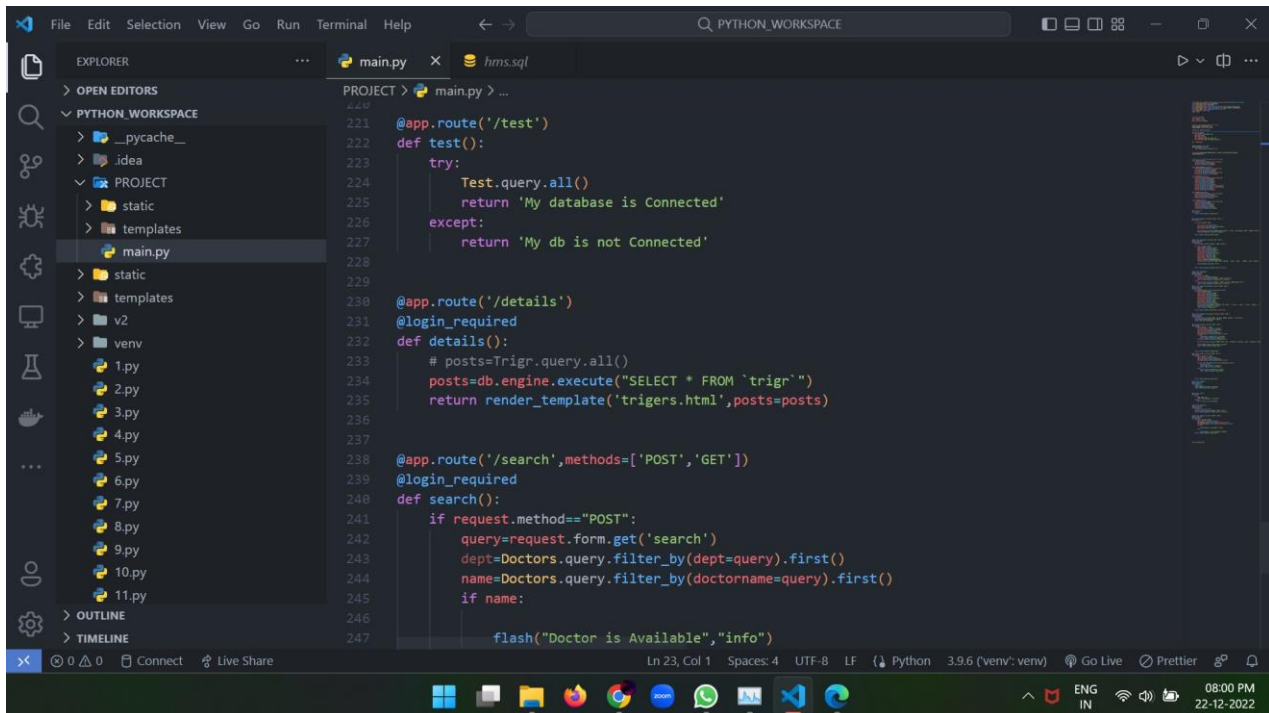
```
161 @app.route("/delete/<string:pid>", methods=['POST', 'GET'])
162 @login_required
163 def delete(pid):
164     db.engine.execute(f"DELETE FROM 'patients' WHERE 'patients'.pid='{pid}'")
165     flash("Slot Deleted Successful", "danger")
166     return redirect('/bookings')
167
168
169 @app.route('/signup', methods=['POST', 'GET'])
170 def signup():
171     if request.method == "POST":
172         username=request.form.get('username')
173         usertype=request.form.get('usertype')
174         email=request.form.get('email')
175         password=request.form.get('password')
176         user=User.query.filter_by(email=email).first()
177         if user:
178             flash("Email Already Exist", "warning")
179             return render_template('/signup.html')
180         encpassword=generate_password_hash(password)
181
182         new_user=db.engine.execute(f"INSERT INTO 'user' ('username','usertype','email','password')
183
184         flash("Signup Succes Please Login", "success")
185         return render_template('login.html')
186
187
188
```

The status bar at the bottom indicates the file is at line 23, column 1, with 4 spaces, UTF-8 encoding, and LF line endings. The system tray shows the date and time as 08:00 PM on 22-12-2022.

The screenshot shows the VS Code editor with the `main.py` file open, displaying the `login` and `logout` endpoints. The main editor area displays the following Python code:

```
188     return render_template('signup.html')
189
190
191 @app.route('/login', methods=['POST', 'GET'])
192 def login():
193     if request.method == "POST":
194         email=request.form.get('email')
195         password=request.form.get('password')
196         user=User.query.filter_by(email=email).first()
197
198         if user and check_password_hash(user.password,password):
199             login_user(user)
200             flash("Login Success", "primary")
201             return redirect(url_for('index'))
202         else:
203             flash("invalid credentials", "danger")
204             return render_template('login.html')
205
206
207
208
209     return render_template('login.html')
210
211
212 @app.route('/logout')
213 @login_required
214 def logout():
215     logout_user()
216
```

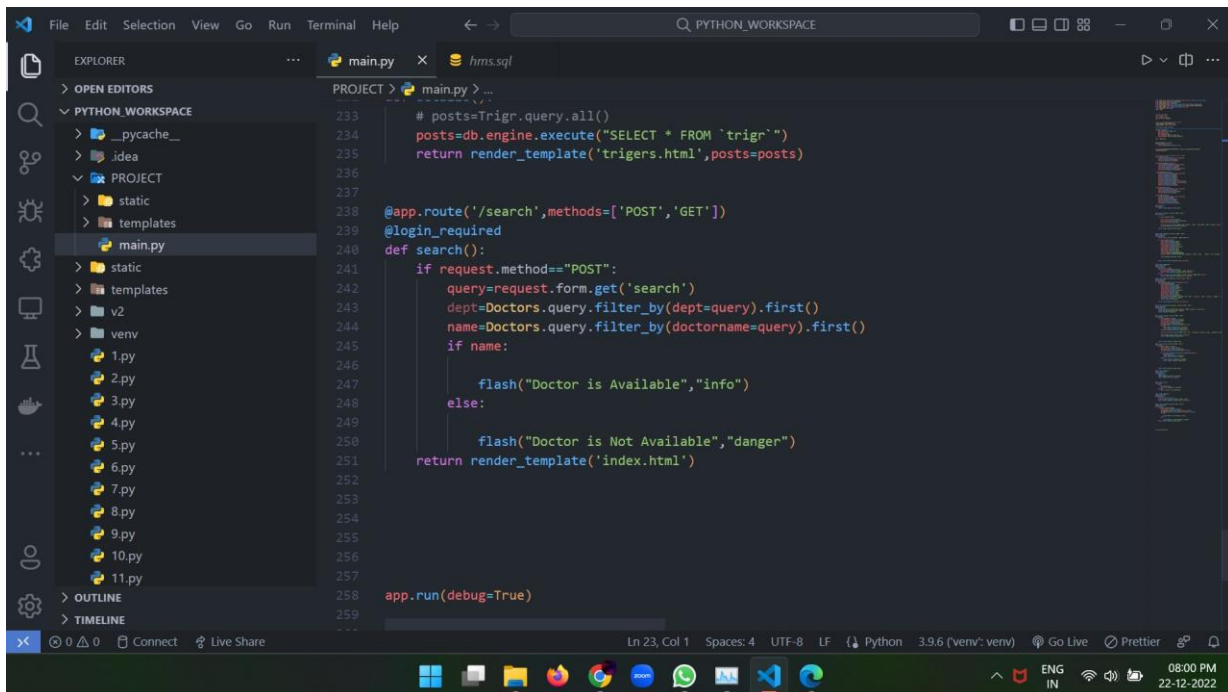
The status bar at the bottom indicates the file is at line 23, column 1, with 4 spaces, UTF-8 encoding, and LF line endings. The system tray shows the date and time as 08:00 PM on 22-12-2022.



This screenshot shows the VS Code editor with a Python workspace. The Explorer panel on the left shows the project structure, including a 'main.py' file. The main editor displays the code for 'main.py', which includes a test route and a details route. The code is as follows:

```
221 @app.route('/test')
222 def test():
223     try:
224         Test.query.all()
225         return 'My database is Connected'
226     except:
227         return 'My db is not Connected'
228
229
230 @app.route('/details')
231 @login_required
232 def details():
233     # posts=Trigr.query.all()
234     posts=db.engine.execute("SELECT * FROM `trigr`")
235     return render_template('triggers.html',posts=posts)
236
237
238 @app.route('/search',methods=['POST','GET'])
239 @login_required
240 def search():
241     if request.method=="POST":
242         query=request.form.get('search')
243         dept=Doctors.query.filter_by(dept=query).first()
244         name=Doctors.query.filter_by(doctorname=query).first()
245         if name:
246
247             flash("Doctor is Available","info")
```

The status bar at the bottom indicates the file is at line 23, column 1, with 4 spaces, in UTF-8 encoding, using the Python 3.9.6 (venv) interpreter. The system tray shows the date and time as 08:00 PM on 22-12-2022.



This screenshot shows the VS Code editor with the same Python workspace. The main editor displays the code for 'main.py', focusing on the search route. The code is as follows:

```
233 # posts=Trigr.query.all()
234 posts=db.engine.execute("SELECT * FROM `trigr`")
235 return render_template('triggers.html',posts=posts)
236
237
238 @app.route('/search',methods=['POST','GET'])
239 @login_required
240 def search():
241     if request.method=="POST":
242         query=request.form.get('search')
243         dept=Doctors.query.filter_by(dept=query).first()
244         name=Doctors.query.filter_by(doctorname=query).first()
245         if name:
246
247             flash("Doctor is Available","info")
248         else:
249
250             flash("Doctor is Not Available","danger")
251     return render_template('index.html')
252
253
254
255
256
257
258 app.run(debug=True)
259
```

The status bar at the bottom indicates the file is at line 23, column 1, with 4 spaces, in UTF-8 encoding, using the Python 3.9.6 (venv) interpreter. The system tray shows the date and time as 08:00 PM on 22-12-2022.


```

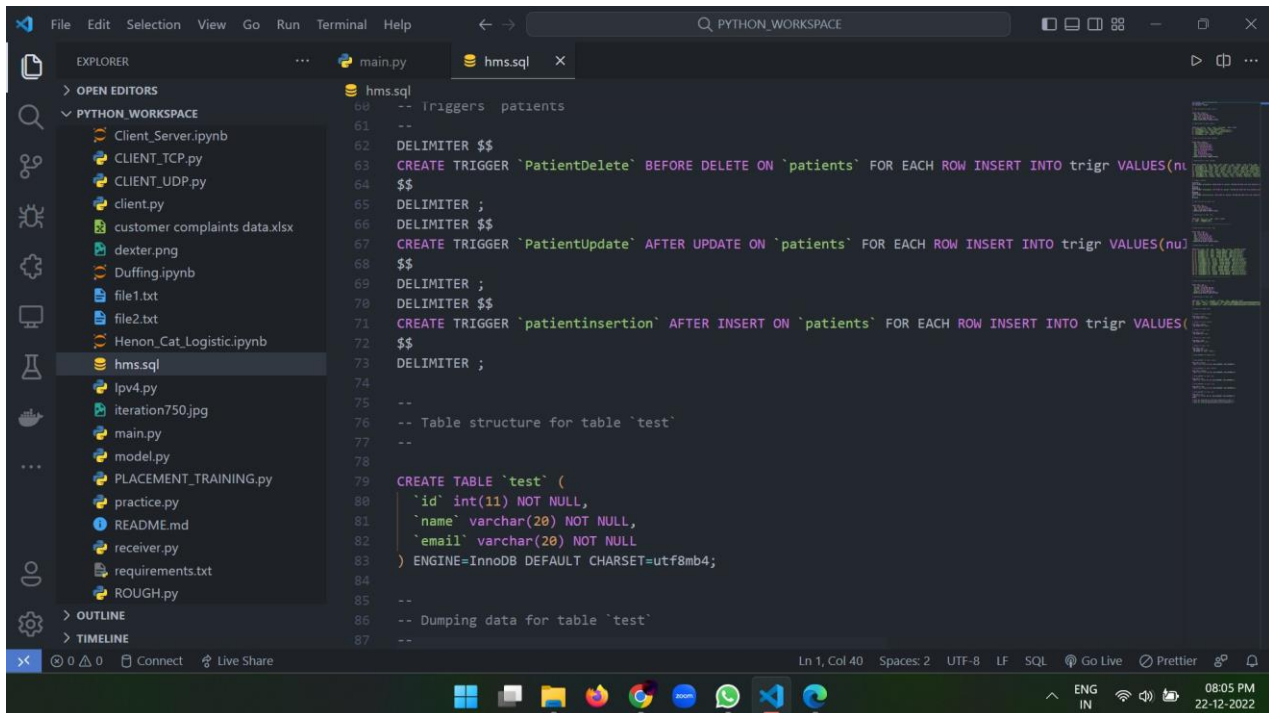
1 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
2 START TRANSACTION;
3 SET time_zone = "+00:00";
4
5 --
6 -- Table structure for table `doctors`
7 --
8
9 CREATE TABLE `doctors` (
10   `did` int(11) NOT NULL,
11   `email` varchar(50) NOT NULL,
12   `doctorname` varchar(50) NOT NULL,
13   `dept` varchar(100) NOT NULL
14 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
15
16 --
17 -- Dumping data for table `doctors`
18 --
19
20 INSERT INTO `doctors` (`did`, `email`, `doctorname`, `dept`) VALUES
21 (1, 'anees@gmail.com', 'anees', 'Cardiologists'),
22 (2, 'amrutha@gmail.com', 'amrutha bhatta', 'Dermatologists'),
23 (3, 'aadithyaa@gmail.com', 'aadithyaa', 'Anesthesiologists'),
24 (4, 'anees@gmail', 'anees', 'Endocrinologists'),
25 (5, 'aneeqah@gmail.com', 'aneekha', 'corona');
26
27 --

```

```

27 --
28 -- Table structure for table `patients`
29 --
30
31 CREATE TABLE `patients` (
32   `pid` int(11) NOT NULL,
33   `email` varchar(50) NOT NULL,
34   `name` varchar(50) NOT NULL,
35   `gender` varchar(50) NOT NULL,
36   `slot` varchar(50) NOT NULL,
37   `disease` varchar(50) NOT NULL,
38   `time` time NOT NULL,
39   `date` date NOT NULL,
40   `dept` varchar(50) NOT NULL,
41   `number` varchar(12) NOT NULL
42 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
43
44 --
45 -- Dumping data for table `patients`
46 --
47
48 INSERT INTO `patients` (`pid`, `email`, `name`, `gender`, `slot`, `disease`, `time`, `date`, `dept`, `number`) VALUES
49 (2, 'anees1@gmail.com', 'anees1 rehman khan', 'Male', 'evening1', 'cold1', '21:20:00', '2020-02-08', 'Cardiologi', '1'),
50 (5, 'patient@gmail.com', 'patien', 'Male', 'morning', 'fevr', '18:06:00', '2020-11-18', 'Cardiologi', '1'),
51 (7, 'patient@gmail.com', 'anees', 'Male', 'evening', 'cold', '22:18:00', '2020-11-05', 'Dermatologi', '1'),
52 (8, 'patient@gmail.com', 'anees', 'Male', 'evening', 'cold', '22:18:00', '2020-11-05', 'Dermatologi', '1'),
53 (9, 'aneesurrehman423@gmail.com', 'anees', 'Male', 'morning', 'cold', '17:27:00', '2020-11-26', 'Ar', '1'),
54 (10, 'anees@gmail.com', 'anees', 'Male', 'evening', 'fever', '16:25:00', '2020-12-09', 'Cardiologi', '1');

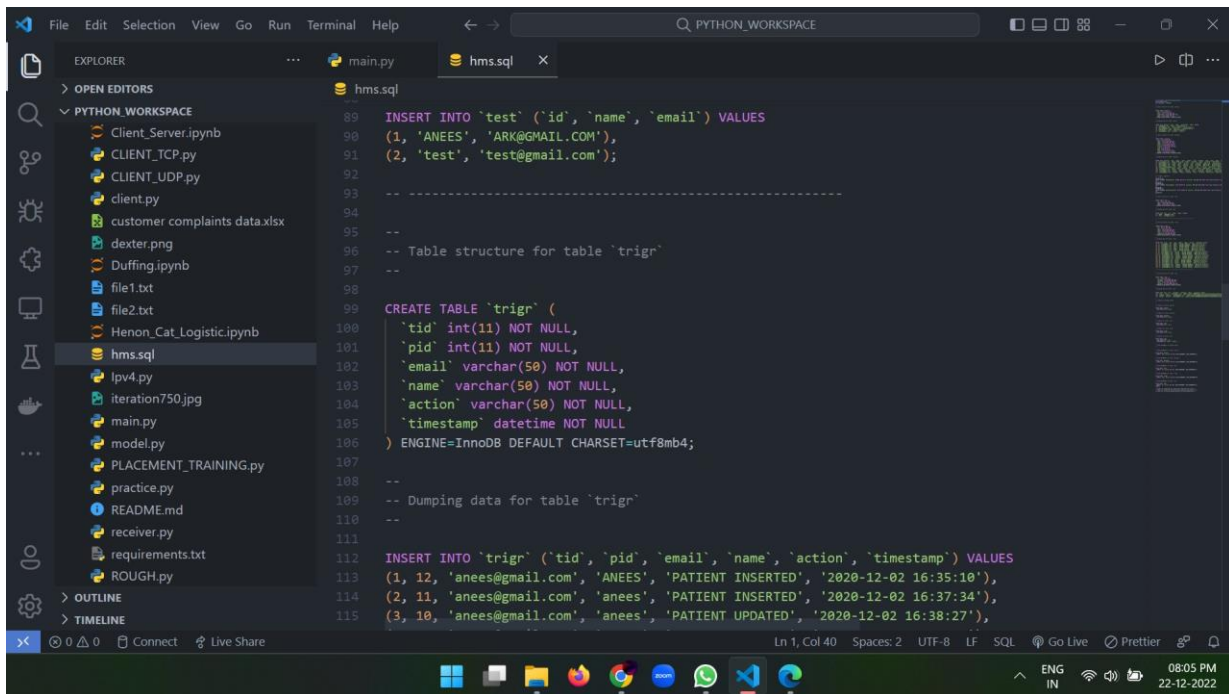
```



VS Code editor window showing the `hms.sql` file. The Explorer sidebar on the left lists files in the `PYTHON_WORKSPACE`, including `Client_Server.ipynb`, `CLIENT_TCP.py`, `CLIENT_UDP.py`, `client.py`, `customer complaints data.xlsx`, `dexter.png`, `Duffing.ipynb`, `file1.txt`, `file2.txt`, `Henon_Cat_Logistic.ipynb`, `hms.sql` (selected), `lpv4.py`, `iteration750.jpg`, `main.py`, `model.py`, `PLACEMENT_TRAINING.py`, `practice.py`, `README.md`, `receiver.py`, `requirements.txt`, and `ROUGH.py`. The main editor displays the following SQL code:

```
60 -- Triggers patients
61 --
62 DELIMITER $$
63 CREATE TRIGGER `PatientDelete` BEFORE DELETE ON `patients` FOR EACH ROW INSERT INTO trigr VALUES(nu
64 $$
65 DELIMITER ;
66 DELIMITER $$
67 CREATE TRIGGER `PatientUpdate` AFTER UPDATE ON `patients` FOR EACH ROW INSERT INTO trigr VALUES(nu
68 $$
69 DELIMITER ;
70 DELIMITER $$
71 CREATE TRIGGER `patientinsertion` AFTER INSERT ON `patients` FOR EACH ROW INSERT INTO trigr VALUES(
72 $$
73 DELIMITER ;
74
75 --
76 -- Table structure for table `test`
77 --
78
79 CREATE TABLE `test` (
80   `id` int(11) NOT NULL,
81   `name` varchar(20) NOT NULL,
82   `email` varchar(20) NOT NULL
83 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
84
85 --
86 -- Dumping data for table `test`
87 --
```

The status bar at the bottom indicates the cursor is at line 1, column 40, with 2 spaces, UTF-8 encoding, and LF line endings. The language is set to SQL. The system tray shows the time as 08:05 PM on 22-12-2022.



VS Code editor window showing the `hms.sql` file. The Explorer sidebar on the left is identical to the previous screenshot. The main editor displays the following SQL code:

```
89 INSERT INTO `test` (`id`, `name`, `email`) VALUES
90 (1, 'ANEES', 'ARK@GMAIL.COM'),
91 (2, 'test', 'test@gmail.com');
92
93 -----
94
95 --
96 -- Table structure for table `trigr`
97 --
98
99 CREATE TABLE `trigr` (
100   `tid` int(11) NOT NULL,
101   `pid` int(11) NOT NULL,
102   `email` varchar(50) NOT NULL,
103   `name` varchar(50) NOT NULL,
104   `action` varchar(50) NOT NULL,
105   `timestamp` datetime NOT NULL
106 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
107
108 --
109 -- Dumping data for table `trigr`
110 --
111
112 INSERT INTO `trigr` (`tid`, `pid`, `email`, `name`, `action`, `timestamp`) VALUES
113 (1, 12, 'anees@gmail.com', 'ANEES', 'PATIENT INSERTED', '2020-12-02 16:35:10'),
114 (2, 11, 'anees@gmail.com', 'anees', 'PATIENT INSERTED', '2020-12-02 16:37:34'),
115 (3, 10, 'anees@gmail.com', 'anees', 'PATIENT UPDATED', '2020-12-02 16:38:27');
```

The status bar at the bottom indicates the cursor is at line 1, column 40, with 2 spaces, UTF-8 encoding, and LF line endings. The language is set to SQL. The system tray shows the time as 08:05 PM on 22-12-2022.

```

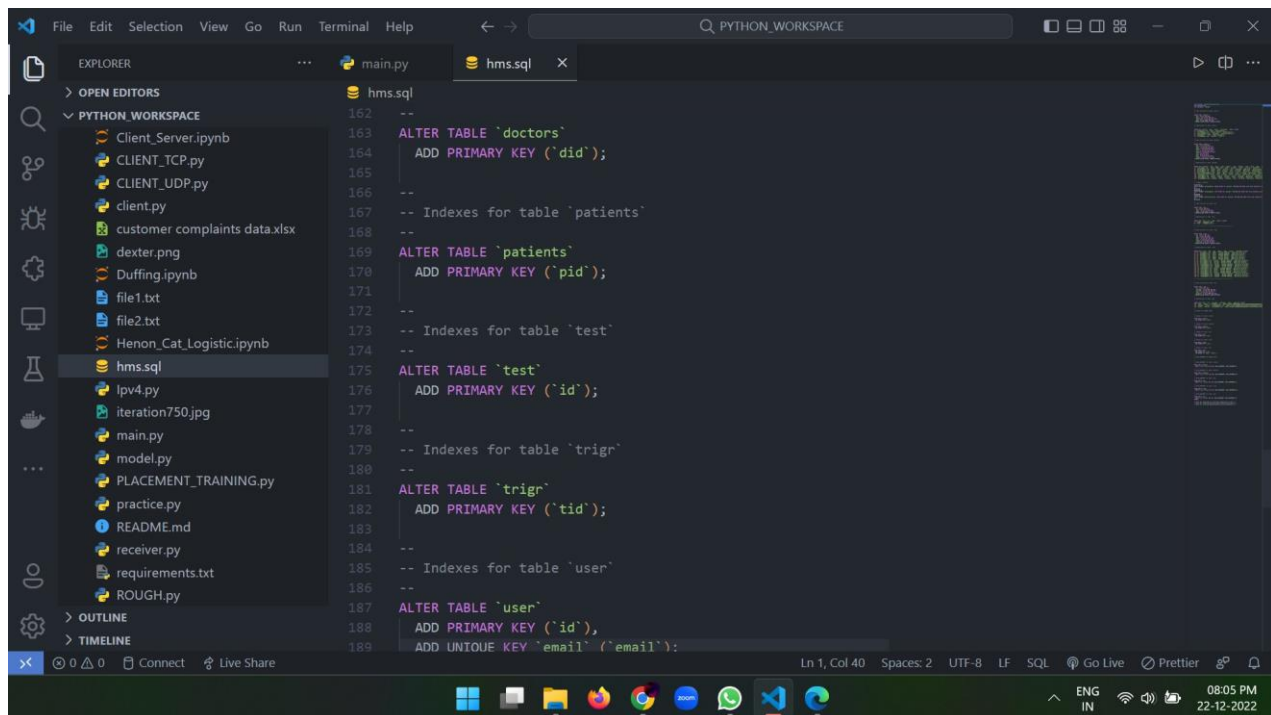
111
112 INSERT INTO `trigr` (`tid`, `pid`, `email`, `name`, `action`, `timestamp`) VALUES
113 (1, 12, 'anees@gmail.com', 'ANEES', 'PATIENT INSERTED', '2020-12-02 16:35:10'),
114 (2, 11, 'anees@gmail.com', 'anees', 'PATIENT INSERTED', '2020-12-02 16:37:34'),
115 (3, 10, 'anees@gmail.com', 'anees', 'PATIENT UPDATED', '2020-12-02 16:38:27'),
116 (4, 11, 'anees@gmail.com', 'anees', 'PATIENT UPDATED', '2020-12-02 16:38:33'),
117 (5, 12, 'anees@gmail.com', 'ANEES', 'Patient Deleted', '2020-12-02 16:40:40'),
118 (6, 11, 'anees@gmail.com', 'anees', 'PATIENT DELETED', '2020-12-02 16:41:10'),
119 (7, 13, 'testing@gmail.com', 'testing', 'PATIENT INSERTED', '2020-12-02 16:50:21'),
120 (8, 13, 'testing@gmail.com', 'testing', 'PATIENT UPDATED', '2020-12-02 16:50:32'),
121 (9, 13, 'testing@gmail.com', 'testing', 'PATIENT DELETED', '2020-12-02 16:50:57'),
122 (10, 14, 'aneeqah@gmail.com', 'aneeqah', 'PATIENT INSERTED', '2021-01-22 15:18:09'),
123 (11, 14, 'aneeqah@gmail.com', 'aneeqah', 'PATIENT UPDATED', '2021-01-22 15:18:29'),
124 (12, 14, 'aneeqah@gmail.com', 'aneeqah', 'PATIENT DELETED', '2021-01-22 15:41:48'),
125 (13, 15, 'khushi@gmail.com', 'khushi', 'PATIENT INSERTED', '2021-01-22 15:43:02'),
126 (14, 15, 'khushi@gmail.com', 'khushi', 'PATIENT UPDATED', '2021-01-22 15:43:11'),
127 (15, 16, 'khushi@gmail.com', 'khushi', 'PATIENT INSERTED', '2021-01-22 15:43:37'),
128 (16, 16, 'khushi@gmail.com', 'khushi', 'PATIENT UPDATED', '2021-01-22 15:43:49'),
129 (17, 17, 'aneeqah@gmail.com', 'aneeqah', 'PATIENT INSERTED', '2021-01-22 15:44:41'),
130 (18, 17, 'aneeqah@gmail.com', 'aneeqah', 'PATIENT UPDATED', '2021-01-22 15:44:52'),
131 (19, 17, 'aneeqah@gmail.com', 'aneeqah', 'PATIENT UPDATED', '2021-01-22 15:44:59');
132
133 --
134 --
135 --
136 -- Table structure for table `user`
137 --
138

```

```

138
139 CREATE TABLE `user` (
140   `id` int(11) NOT NULL,
141   `username` varchar(50) NOT NULL,
142   `usertype` varchar(50) NOT NULL,
143   `email` varchar(50) NOT NULL,
144   `password` varchar(1000) NOT NULL
145 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
146
147 --
148 -- Dumping data for table `user`
149 --
150
151 INSERT INTO `user` (`id`, `username`, `usertype`, `email`, `password`) VALUES
152 (13, 'anees', 'Doctor', 'anees@gmail.com', 'pbkdf2:sha256:150000$xAkZC1jG$4c7a7e704708f86659d73056'),
153 (14, 'aneeqah', 'Patient', 'aneeqah@gmail.com', 'pbkdf2:sha256:150000$Yf511lDC$028cff81a536ed9d477f'),
154 (15, 'khushi', 'Patient', 'khushi@gmail.com', 'pbkdf2:sha256:150000$BeShRKv$a8b27379ce9b2499d4cae');
155
156 --
157 -- Indexes for dumped tables
158 --
159
160 --
161 -- Indexes for table `doctors`
162 --
163
164 ALTER TABLE `doctors`
165   ADD PRIMARY KEY (`did`);
166

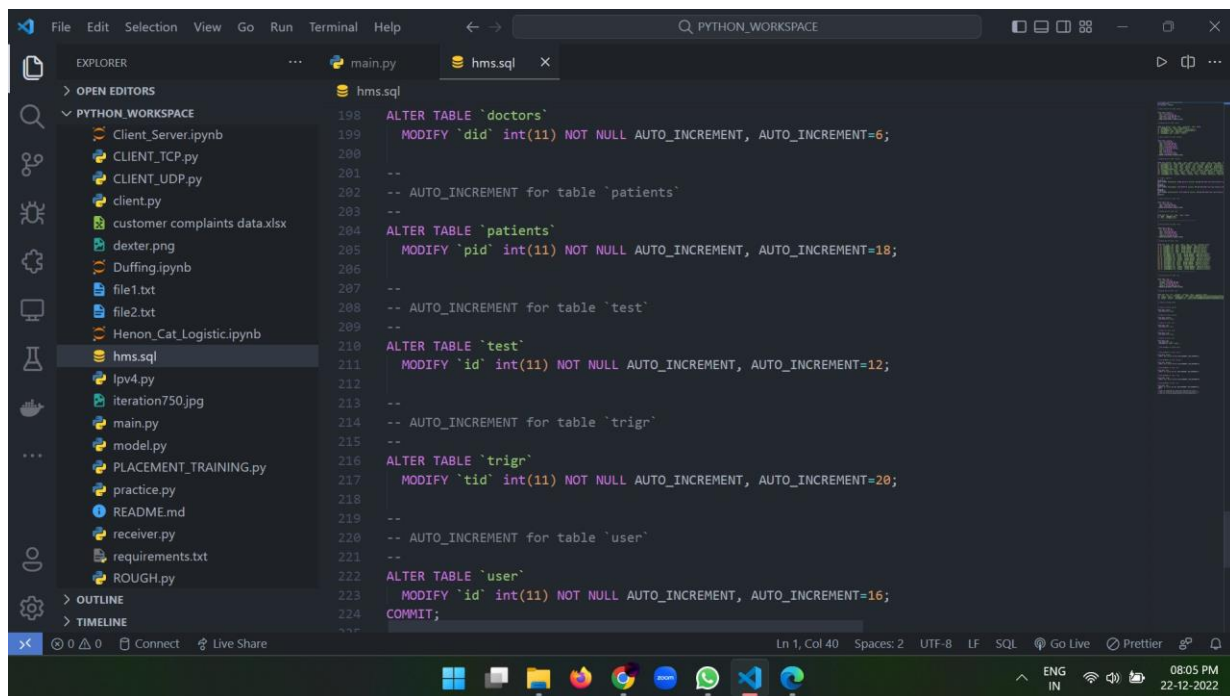
```

This screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'PYTHON_WORKSPACE'. The file 'hms.sql' is selected and open in the editor. The code in the editor is as follows:

```
162 --
163 ALTER TABLE `doctors`
164     ADD PRIMARY KEY (`did`);
165 --
166 -- Indexes for table `patients`
167 --
168 --
169 ALTER TABLE `patients`
170     ADD PRIMARY KEY (`pid`);
171 --
172 -- Indexes for table `test`
173 --
174 --
175 ALTER TABLE `test`
176     ADD PRIMARY KEY (`id`);
177 --
178 -- Indexes for table `trigr`
179 --
180 --
181 ALTER TABLE `trigr`
182     ADD PRIMARY KEY (`tid`);
183 --
184 -- Indexes for table `user`
185 --
186 --
187 ALTER TABLE `user`
188     ADD PRIMARY KEY (`id`),
189     ADD UNIQUE KEY `email` (`email`);
```

The status bar at the bottom indicates the cursor is at line 1, column 40, with 2 spaces, UTF-8 encoding, and LF line endings. The language is set to SQL. The system tray shows the time as 08:05 PM on 22-12-2022.



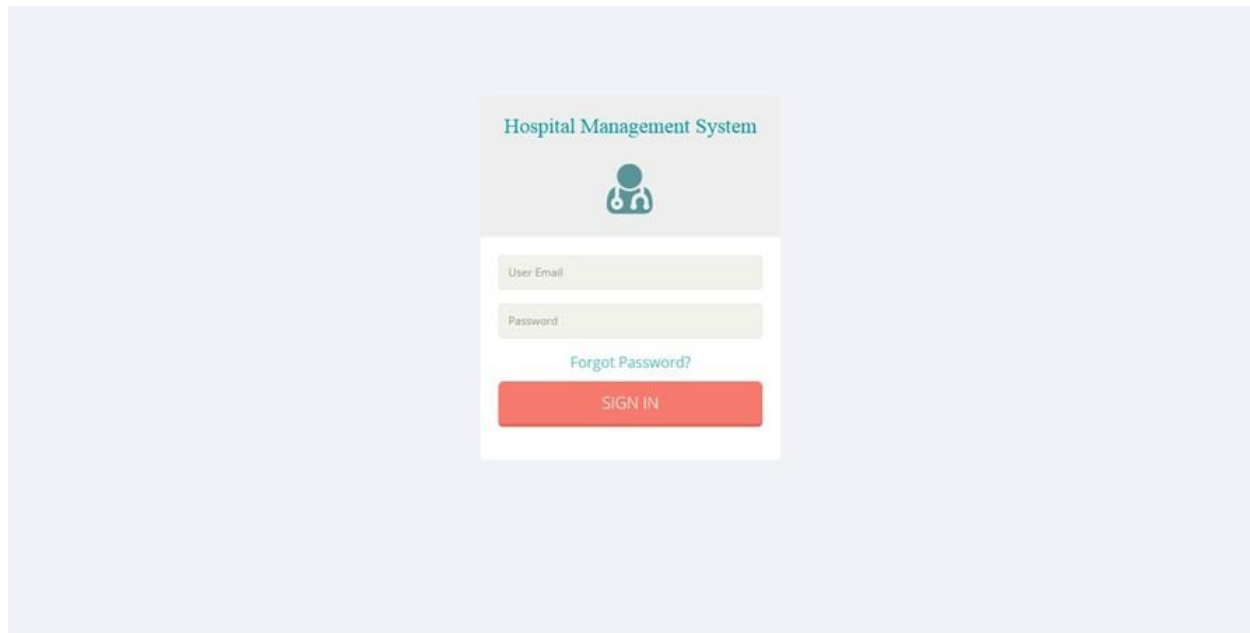
This screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'PYTHON_WORKSPACE'. The file 'hms.sql' is selected and open in the editor. The code in the editor is as follows:

```
198 ALTER TABLE `doctors`
199     MODIFY `did` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
200 --
201 -- AUTO_INCREMENT for table `patients`
202 --
203 --
204 ALTER TABLE `patients`
205     MODIFY `pid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;
206 --
207 -- AUTO_INCREMENT for table `test`
208 --
209 --
210 ALTER TABLE `test`
211     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
212 --
213 -- AUTO_INCREMENT for table `trigr`
214 --
215 --
216 ALTER TABLE `trigr`
217     MODIFY `tid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=20;
218 --
219 -- AUTO_INCREMENT for table `user`
220 --
221 --
222 ALTER TABLE `user`
223     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;
224 COMMIT;
```

The status bar at the bottom indicates the cursor is at line 1, column 40, with 2 spaces, UTF-8 encoding, and LF line endings. The language is set to SQL. The system tray shows the time as 08:05 PM on 22-12-2022.

Site Snapshots

LOGIN PAGE



The login page features a light gray background. In the center, there is a white rectangular box containing the following elements: the title "Hospital Management System" in a dark teal font, a teal icon of a person with a stethoscope, a "User Email" input field, a "Password" input field, a "Forgot Password?" link in teal, and a red "SIGN IN" button.

MAIN PAGE



DEPARTMENTS

The screenshot shows the 'Department' management page. On the left is a dark sidebar with navigation links: Dashboard, Departments (active), Doctor, Patient, Human Resources, Financial Activities, Medicine, Donor, Bed, Report, Settings, and Profile. The main content area has a teal header 'Department' and a '+ Add Department' button. Below this is a search bar and a table with columns 'Name', 'Description', and 'Options'. The table contains two entries: 'Surgery' and 'Cardiology'. At the bottom, it says 'Showing 1 to 2 of 2 entries' with 'Prev' and 'Next' navigation buttons.

Name	Description	Options
Surgery	Is a branch of medicine that deals with physical structure to diagnose or prevent treatment	
Cardiology	Is a branch of internal medicine concerned with heart health and treatment of the heart and the circulatory system.	

NEW EMPLOYEE DOCTOR

The screenshot shows the 'Add Doctor' form. The sidebar is the same as the previous page. The main content area has a teal header 'Doctors' and a '+ Add Doctor' button. The form fields are: Name, Password (masked with asterisks), Email, Address, Phone, Department (a dropdown menu currently showing 'Cardiology'), Profile, and Image (with a file upload button labeled 'اختر ملف'). At the bottom, it says 'Showing 1 to 3 of 3 entries' with 'Prev' and 'Next' navigation buttons.

DOCTOR MODULE

1

0

0

0

0

0

Hospital Management System

Doctors

Add Doctor

Print

All

 records per page

Search:

Image	Name	Email	Address	Phone	Department	Profile	Options
	Dr.Abdulrahman	Abdulrahman@example.com	ksa,ryiad	+966552476030	Diabetes	?????? ???? ???? ?	<div><div></div><div></div></div>
	Dr.Wael	Wael@example.com	ksa,ryiad	+966552450055	Surgery	?????? ???? ?	<div><div></div><div></div></div>
	Dr.Ahmad	Dr.Ahmad@example.com	ksa,ryiad	+96609123123123	Cardiology	?????? ???? ???? ? ?	<div><div></div><div></div></div>

OPERATION DETAILS

1

0

0

0

0

0

Hospital Management System

Patient

Add New Patient

Print

All

 records per page

Search:

Patient Id	Image	Name	Email	Doctor	Birth Date	Phone	Blood Group	Due Balance	Options
292848		norra	norra@example.com	Dr.Wael	03-01-2012	+966552476030	AB-	SAR 0	<div><div></div> details</div> <div>Invoice</div>
921383		Mr.patient	patient@example.com	Dr.Abdulrahman	12-01-2015	+966552457380	A+	SAR 54945	<div><div></div> details</div> <div>Invoice</div>

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

</

Hospital Management System

Alloted Beds

Add Allotment
Print

All records per page
Search:

Bed Id	Patient	Alloted Time	Discharge Time	Options
Small-22	Mr Patient	14 November 2018 - 02:15 PM	21 November 2018 - 02:15 PM	

Showing 1 to 1 of 1 entries

Prev
Next

PAYMENT DETAILS

Hospital Management System

Payments

Add Payment
Print

All records per page
Search:

Patient	Date	Sub Total	Discount	vat	Total	Status	Options
norra Ksa,alzulfi +966552476030	11/14/2018	SAR 10000	SAR 50%	SAR 4975	SAR 14925	Paid	Edit Invoice
Mr.patient ksa,ryiad +966552457380	11/14/2018	SAR 50000	SAR 50%	SAR 4995	SAR 54945	Unpaid	Edit Invoice

BIRTH REPORT

Hospital Management System

Birth Report

Add New Report
Print

All records per page
Search:

Patient	Description	Doctor	Date	Options
Naser	??????	Dr,Ahmad	03-06-2018	Edit
Mr Patient	c a r	Mr Doctor	11-30-2015	Edit

Showing 1 to 2 of 2 entries
← Prev
Next →

OPERATION REPORT

Hospital Management System

Operation Report

Add New Report
Print

All records per page
Search:

Patient	Description	Doctor	Date	Options
Naser	Operation in hand	Dr.Wael	03-07-2018	Edit
Mr Patient	Knee surgery	Dr.Wael	03-07-2018	Edit