



# Health Records Management System

Husam Alanazi

Jack Saunders

Abdullah Alshamrani

Jake MEDLOCK

69yo M 08-27-1951 #2446

Patient status ready for staff

Patient location Waiting Room - Jake Medlock

Done with Intake

## Follow-up

- ☒ Vitals
- ☒ Allergies
- ☒ Medications
- ☒ Vaccines
- ☒ Problems
- ☒ Family History
- ☒ Social History
- ☒ Surgical History
- ☒ Past Medical History
- ☒ Screening
- ☒ Clinical Add-Ons
- ☒ Quality Measures (7)
- ☒ History of Present Illness
- ☒ Review of Systems

## Vitals

Next

Ht	6	ft		in		6 ft on 10-02-2020	+
<input type="checkbox"/> Not Performed <a href="#">-select reason-</a>							
Wt	190	lbs	With clothes			190 lbs on 10-02-2020	+
<input type="checkbox"/> Not Performed <a href="#">-select reason-</a>							
<input type="checkbox"/> Out of Range							
BMI	25.8					25.8 on 10-02-2020	
BP	136	/	84		sitting	136 / 84 on 10-02-2020	+
<input type="checkbox"/> Not Performed <a href="#">-select reason-</a>							
Pulse	64	bpm				64 bpm on 10-02-2020	+
RR	16					16 on 10-02-2020	+
O2Sat	98	%				98 % on 08-25-2020	+
T		°F				98.6 °F on 10-02-2020	+

# How it works?

Abdul

Client

Server  
{Node.JS}

Database  
MySQL

- (1) User issues URL from a browser  
http://host:port/path/file



- (5) Browser formats the response  
and displays

Client (Browser)

- (2) Browser sends a request message

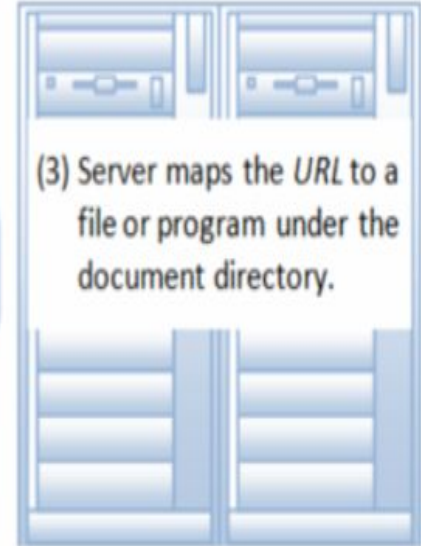
GET URL HTTP/1.1  
Host: host:port  
.....  
.....

- (4) Server returns a response message

HTTP/1.1 200 OK  
.....  
.....  
.....

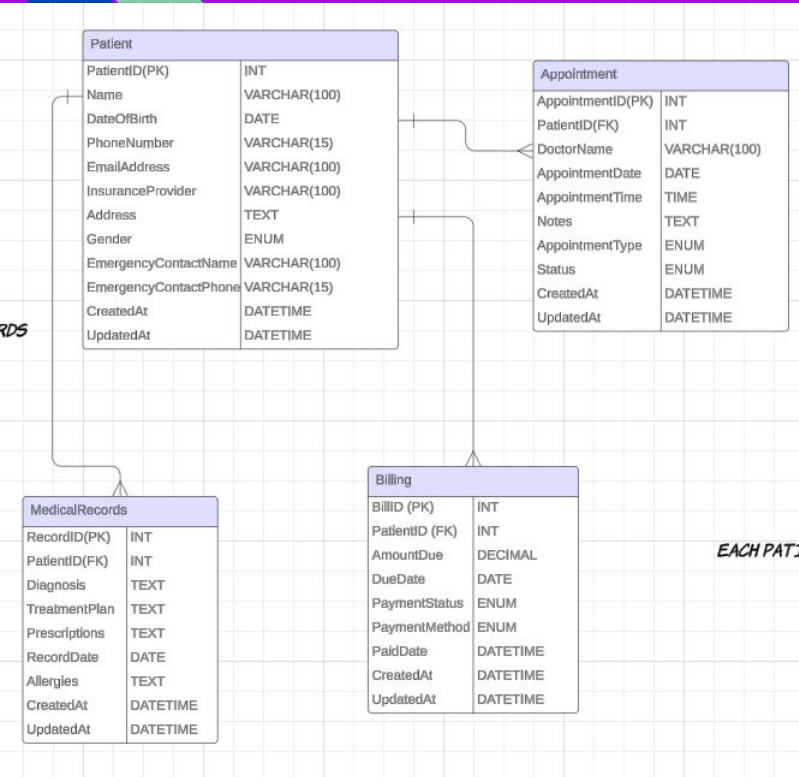
HTTP (Over TCP/IP)

- (3) Server maps the URL to a  
file or program under the  
document directory.



Server (@ host:port)

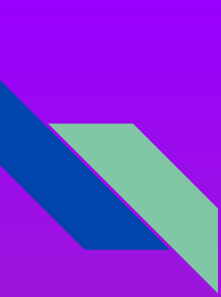
# ER Diagram and Database {MySQL} Structure



```
-- Create Patients table
CREATE TABLE Patients (
    PatientID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    DateOfBirth DATE NOT NULL,
    PhoneNumber VARCHAR(15) NOT NULL,
    EmailAddress VARCHAR(100),
    InsuranceProvider VARCHAR(100),
    Address TEXT,
    Gender ENUM('Male', 'Female', 'Other'),
    EmergencyContactName VARCHAR(100),
    EmergencyContactPhone VARCHAR(15),
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

```
INSERT INTO Patients (Name, DateOfBirth, PhoneNumber, EmailAddress, InsuranceProvider, Gender)
VALUES ('John Doe', '1990-01-01', '1234567890', 'johndoe@example.com', 'ABC Insurance', 'Male');
```

```
CREATE INDEX idx_patient_name ON Patients(Name);
CREATE INDEX idx_patient_dob ON Patients(DateOfBirth);
```



```
UPDATE Patients
```

```
SET
```

```
    PhoneNumber = '0987654321',
```

```
    EmailAddress = 'john.doe_updated@example.com',
```

```
    Address = '456 Elm St',
```

```
    UpdatedAt = NOW()
```

```
WHERE
```

```
    PatientID = 1;
```

```
SELECT *
```

```
FROM Patients
```

```
WHERE
```

```
    Name LIKE '%John%'
```

```
    AND DateOfBirth = '1990-01-01';
```

# Web Page {HTML, CSS, JavaScript}

Jack

```
10 background-color: #befdbd; /* Set
11 color: white; /* Set text color t
12 font-family: Arial, sans-serif; /
13 }
14 /* Style for buttons */
15 .button {
16 border: none;
17 color: white;
18 padding: 15px 32px;
19 text-align: center;
20 text-decoration: none;
21 display: inline-block;
22 font-size: 19px;
23 margin: 4px 2px;
24 cursor: pointer;
25 position: relative;
26 }
```

```
<div class="horizontal-center">
  <button class="button TL" id="roundCornersTL" onclick="window.location.href='PAPatients.html'">
    <h1>Patients</h1>
    <p>View or edit the hospital's patients</p>
  </button>
```

```
<div align="center">
  <INPUT TYPE="submit" VALUE="Add">
  <INPUT TYPE="submit" VALUE="Delete">
  <INPUT TYPE="submit" VALUE="Search">
  <INPUT TYPE="submit" VALUE="Update">
</div>
<BR>
<div align="center">
  <INPUT TYPE="reset" VALUE="Reset">
</div>
```

# Node.JS

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
const dotenv = require('dotenv');

const patientRoutes = require('./routes/patientRoutes');
const { errorHandler } = require('./middleware/errorMiddleware');

dotenv.config();

const app = express();

// Middleware
app.use(cors());
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// Routes
app.use('/api/patients', patientRoutes);
app.use('/api/patients', require('./routes/patientRoutes'));

// Error Handling Middleware
app.use(errorHandler);

// Start Server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

```
// Add Patient Controller
exports.addPatient = async (req, res, next) => {
  try {
    const { name, age, medicalHistory } = req.body;

    // Validate input data
    if (!name || !age || !medicalHistory) {
      return res.status(400).json({ message: 'All fields are required' });
    }

    // Insert patient into the database
    const [result] = await db.execute(
      'INSERT INTO patients (name, age, medical_history) VALUES (?, ?, ?)',
      [name, age, medicalHistory]
    );

    // Send success response
    res.status(201).json({
      message: 'Patient added successfully',
      patientId: result.insertId,
    });
  } catch (error) {
    next(error); // Pass errors to middleware
  }
};
```

# Routing Node.js to MySQL

Node.js routes requests from the user to MySQL using specific endpoints

```
const express = require('express');  
const { addPatient, getPatients } =  
  require('./controllers/patientController');  
const router = express.Router();
```

```
// Route to add a patient  
router.post('/', addPatient);
```

```
// Route to fetch patients  
router.get('/', getPatients);
```

```
module.exports = router;
```

```
// Add Patient Controller  
exports.addPatient = async (req, res, next) => {  
  try {  
    const {  
      name,  
      dateOfBirth,  
      phoneNumber,  
      emailAddress,  
      insuranceProvider,  
      gender,  
      address,  
      emergencyContactName,  
      emergencyContactPhone,  
    } = req.body;  
  
    // Validate input data  
    if (  
      !name ||  
      !dateOfBirth ||  
      !phoneNumber ||  
      !emailAddress ||  
      !insuranceProvider ||  
      !gender  
    ) {  
      return res.status(400).json({ message: 'All fields are required' });  
    }  
  }  
}
```



# Routing Node.Js to the Web Page

```
<FORM NAME="home" ACTION="http://localhost:5000/PAhome" METHOD="post">
</FORM>
```

```
<FORM NAME="Main" ACTION="http://localhost:5000/PABilling" METHOD="post">
```

```
<FORM NAME="Main" ACTION="http://localhost:5000/PAREcords" METHOD="post">
```

```
<FORM NAME="Main" ACTION="http://localhost:5000/PAPatients" METHOD="post">
```

```
<FORM NAME="Main" ACTION="http://localhost:5000/PAppointments" METHOD="post">
```

```
Patient ID:
<INPUT TYPE="TEXT" NAME="pid" VALUE="" >
<br><br>

Doctor Name:
<INPUT TYPE="TEXT" NAME="DName" VALUE="" >
<br><br>

Appointment Date:
<INPUT TYPE="TEXT" NAME="date" VALUE="" >
<br><br>

Appointment Time:
<INPUT TYPE="TEXT" NAME="time" VALUE="" >
<br><br>

Appointment Type:
<SELECT NAME="AType">
  <OPTION SELECTED>Checkup</OPTION>
  <OPTION>Follow-up</OPTION>
  <OPTION>Consultation</OPTION>
</SELECT>
<br><br>

Appointment Status:
<SELECT NAME="status">
  <OPTION SELECTED>Scheduled</OPTION>
  <OPTION>Completed</OPTION>
  <OPTION>Cancelled</OPTION>
</SELECT>
```

## Appointments

Patient ID:

Doctor Name:

Appointment Date:

Appointment Time:

Notes:

Appointment Type:

Appointment Status:

[Go Home](#)





THANK  
YOU