ELP - 720 Telecom Networks Laboratory

Anshuman Singh
2018JTM2004
2018-19

A report presented on

Wireless Ordering & Billing System for Restaurants

Bharti School
of
Telecommunication Technology and Management
IIT DELHI, Delhi

April 29, 2019

Contents

List of Figures

1 Introduction

ℹ

Problem Statement Many times in hotel we have to wait for a waiter to give our order for food. This creates problem when there is rush in hotel especially in festival seasons and generally on weekends. Also, there are self-service hotels where customers have to get up each time to place and receive their orders. Customers also expect to have an idea about service time and total bill beforehand. Main intention of our project is to avoid such problems and to give a cost effective solution to such problems.

ℹ

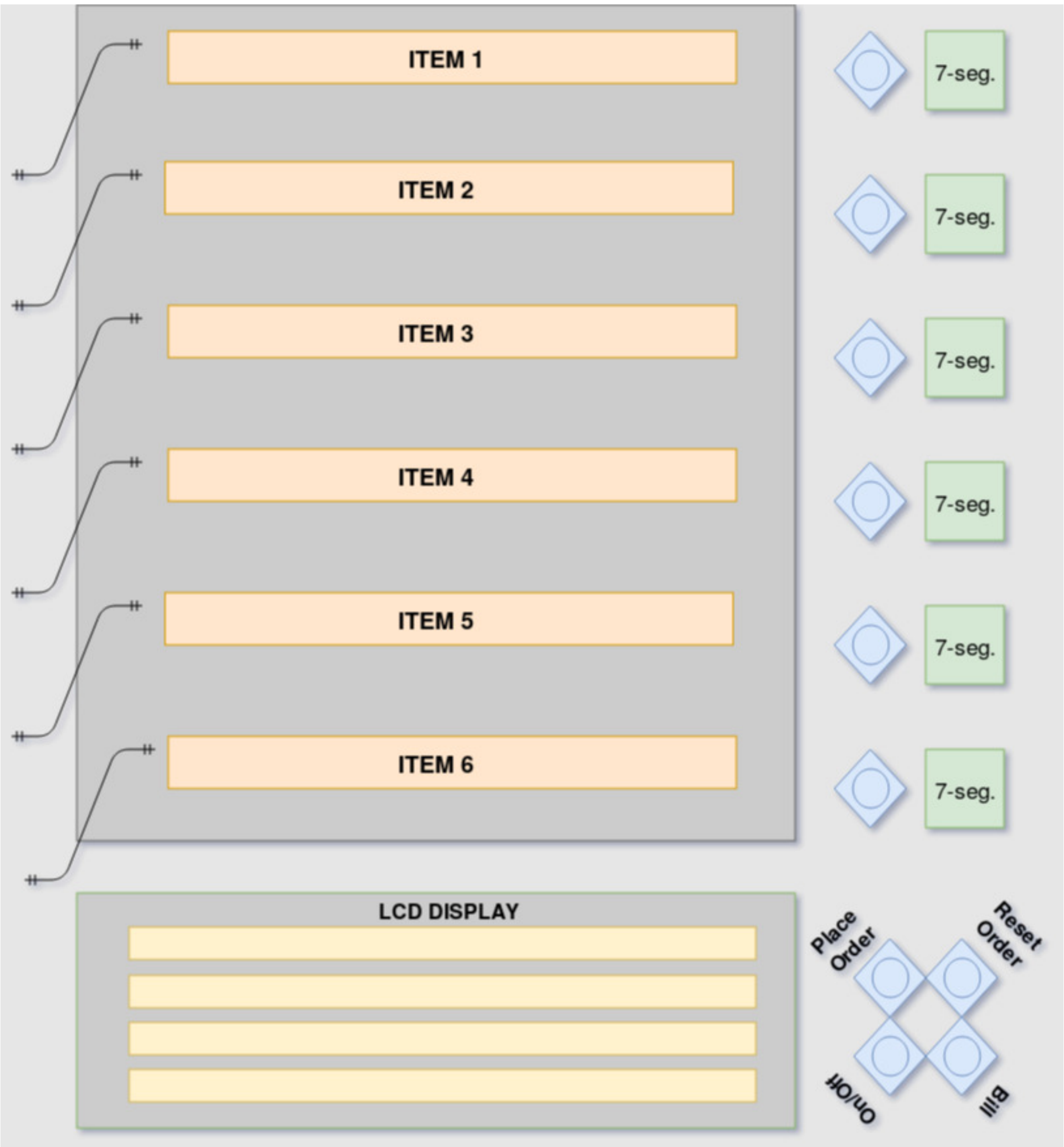
Proposed Solution The project is based on client-server model where the customer sitting at a table is the client and the chef or employees of the restaurant will act as server. The user will be presented with a paper based menu along with a few buttons and an LCD panel. The paper based menu will be modular in terms of Menu Card attached which can be changed/updated by just changing the card. Next to menu items there will be buttons to choose an item on a particular page. Since items are distributed on a number of pages and only few buttons are available to make selections, there will a mechanism available to determine which page user is on. Depending on the page opened, pressing a button next to a menu item will select the item on that particular page and show the same on LCD below. A seven segment display can be attached next to each Button to display count of selections made corresponding to each page. After choosing the menu items order can be placed using a button. This order will be sent through Wi-Fi to the server which is a computer running a server. The Server side will do all the processing on order and send an appropriate message which will be displayed on LCD. At the end when user requests the Bill the same can be displayed on LCD.

2 Project Architecture

In total, project consist of a micro-controller (ESP32), a 20*4 LCD, few buttons, 7 segment displays, LDRs, resistors and a paper menu card all put together on a single board. The menu card has a predefined format in which different menu items are listed in different rows. The menu items on a single page could vary from 8 to greater than 50. In between the pages LDRs have been embedded. For every page turned on, a different set of LDRs get exposed to light. To keep the detection of page accurate paper chosen be thick and dark-coloured. Also, an extra LDR has been placed outside the menu card to act as reference for different lightening conditions. Next to each menu item a button is available to select the corresponding item.

Below the menu card exists a 20*4 LCD display to show each input given and response received from server. Next to the LCD, exists button to configure the order like placing an order, resetting the selections made and making requests for bill. Users can turn through the pages, choose an item, select it and place an order for the same. Since there are limited buttons and many menu items arranged on different pages LDR circuit determines which page user is on. All of the above items are connected with different GPIOs of ESP-32 which is a wifi based micro-controller.

Figure 1: Proposed Layout of Project



ESP32 acts as a client for a python server running on a computer. For every input to the system it sends corresponding requests to server. The communication between the client and server is through a TCP socket. Server running on computer when turned on starts listening for incoming connections from different tables each with its own menu card. Unique identification ID of each ESP32 helps to identify the incoming request. The server makes use of MySQL database to store the request, response, details of every item available along with price and quantity. For different restaurants a different configuration file can be loaded into the MySQL database.

3 Implementation (Tools and Technologies)

The implementation of the project can be divided into the following parts

Socket based Client Server Architecture Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other devices using standard Unix file descriptors. Although different types of sockets are available like Stream Sockets, Datagram Sockets, Raw Sockets etc., Stream socket has been used to establish a connection with server as delivery in a networked environment is guaranteed. They use TCP (Transmission Control Protocol) for data transmission. If delivery is impossible, the sender receives an error indicator. Data records do not have any boundaries.

ESP32 when turned on establishes a connection with the pre-fixed network using its SSID and passphrase. Once connected to network it establishes a socket connection with the server using server's IP address and port number.

I2C protocol to communicate with LCD An I2C protocol is one of the serial communication protocol that is used for the chip to chip communication. The I2C is the short form of Inter-Integrated Circuit, is a type of bus, which designed and developed by Philips in 1980 for inter-chip communication. I2C is basically two-wire communication protocol.It uses only two wire for the communication. In which one wire is used for the data (SDA) and other wire is used for the clock (SCL).

As the number of GPIOs on ESP32 are limited an I2C adapter has been used to communicate with the LCD display with only two pins namely SDA (Data) and SCL (Clock).

Voltage Divider Circuit A voltage divider circuit has been made using resistors. This voltage value is then read using an analog pin of ESP32. When a button is pressed the voltage drop changes. This change in analog pin reading determines which button is pressed. Four buttons can be safely placed on one analog with a sufficient margin to avoid any errors.

Light Dependent Resistors LDR is a component whose light intensity changes as light falls upon it. This property of LDR has been used here in the project. As the intensity of light falling on LDR changes the resistance offered by the LDR changes. Higher the ight intensity falling on LDR lesser the resistance it offers.

LDR's are placed withing the menu card in a way that at each page turn a different set of LDR's get exposed.

4 Implementation

The entire project can be broadly divided into the following sections:

> **Determining Page Number** As the menu items are distributed over a few pages and only limited buttons are available to the user, a mechanism based on LDR's to determine the page number is proposed. LDR is a component whose light intensity changes as light falls upon it. This property of LDR has been used here in the project. In the proposed model 4 LDR's have been used in total. One LDR is always exposed to light and determines the current lightening conditions. Three other LDR's are embedded in the menu and a different set of LDR gets exposed as pages are turned. Readings of all these LDR are then compared with the reference LDR to find the page number.

> **Getting user input from buttons** There are limited number of GPIO's in ESP32. However a minimum of 10 user inputs for different options needs to be taken. To solve this problem analog pins have been used to get status of buttons. The buttons are connected with a voltage divider circuit with the help of resistors. When a button is pressed the potenial drop changes and this is read through the analog pins of ESp32. A total of 4 buttons have been connected to a single analog pins. This way only 3 analog pins have used to get inputs from all the buttons.

> **Displaying User Input and Server Response** To make the interface available to the user more intuitive, a 7 segment display has been placed next to each button which displays the number of items ordered for that menu item. As the page is turned this display gets updated to show the selections made in that page. Also, a 20*4 LCD has been placed at the bottom of the menu to display the selections made and response from the server. Out of the 4 rows the first row shows the response from server. The middle two rows shows the selection made and user input. The last displays the current status of selections. This gets dynamically updated after a particular time instant as specified in the code.

> **Connecting with Server** Server in this project is a Python Socket Server running on a computer and client is the ESP32 placed on the menu card. When the menu card is turned on ESP32 establishes a connection with Server after connecting to the same network. Here, Server uses a socket connection to communicate with the ESP32 using TCP protocol. For every selection made a request is sent to the server and the response received is shown on the LCD's 1st row.

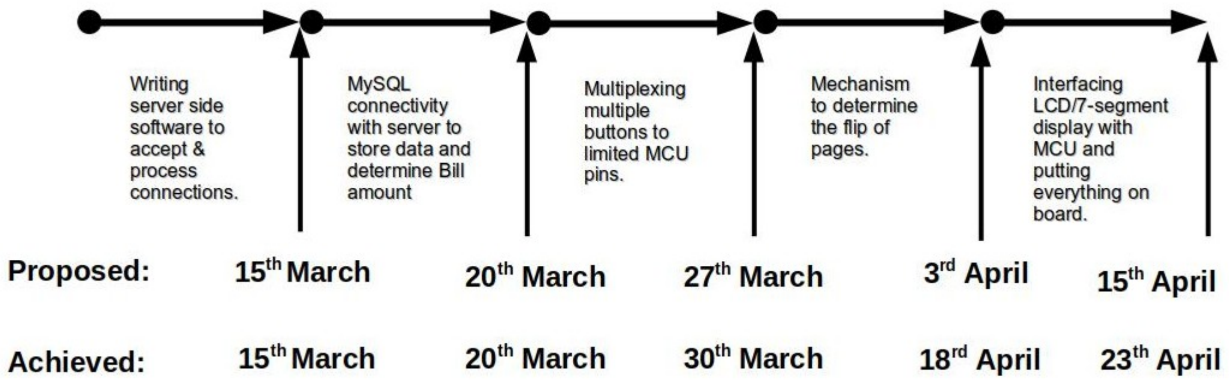
Storing Data and Configuration To store the data about available items, past orders, current orders and configuration files MySQL database server has been used. When menu card has to be changed a new configuration file specifying page number, item number and item id can be sourced into MySQL database. To keep records a total of 5 tables in SQL have been used. One table named "available" keeps record of available items, quantity and price uniquely identifiable by their item id's. Next, a table "config" is used for configuring menu card. To keep selections made by ddifferent tables another table named "temp" is used. Another two tables namely "current orders" and "past orders" maintains record of orders placed and billed.

5 Bill of Materials

Item Name	Quantity	Price (Rs.)
ESP32	1	480
LCD 20*4	1	400
7 segment	6	54
LDRs	4	32
Others	-	50
Total	–	**1016**

6 Proposed Milestones and timeline of milestones achieved

Figure 2: Proposed Milestones and timeline of milestones achieved

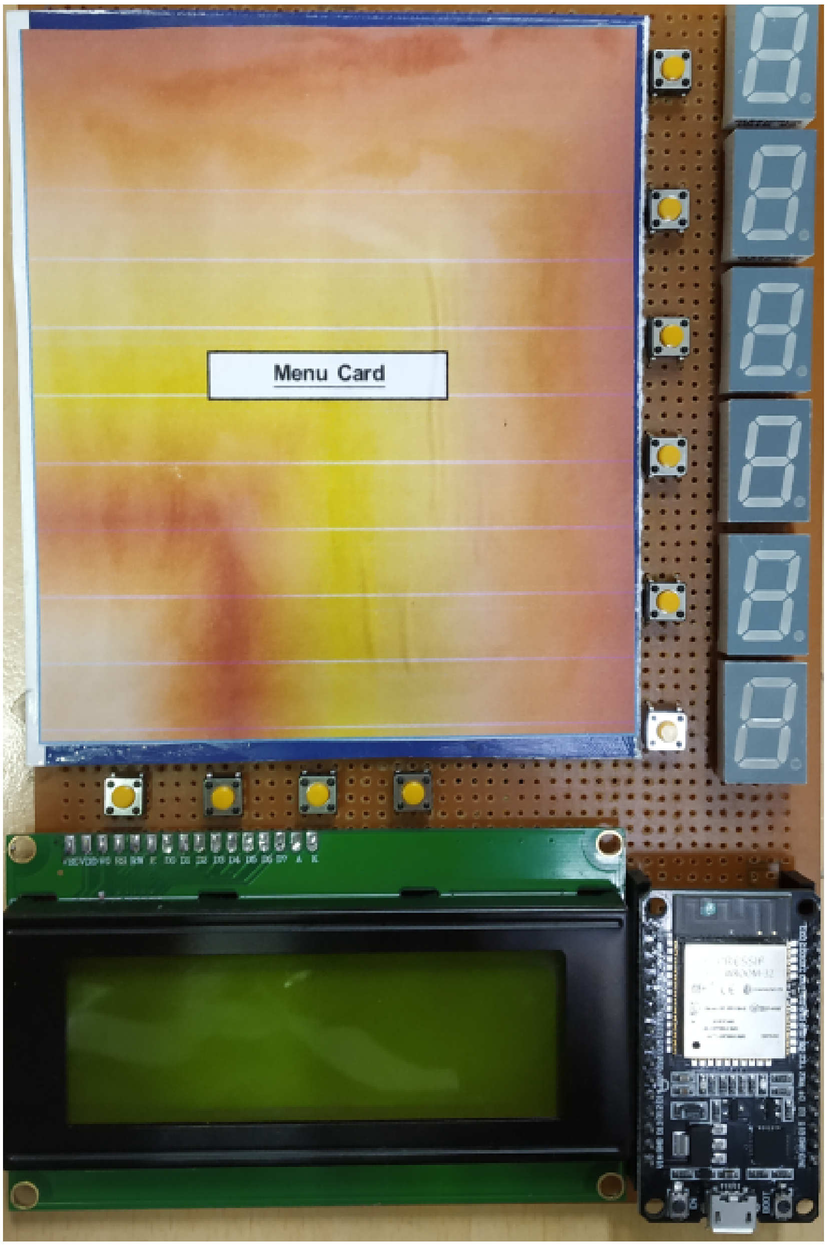

Writing server side software to accept & process connections.
MySQL connectivity with server to store data and determine Bill amount
Multiplexing multiple buttons to limited MCU pins.
Mechanism to determine the flip of pages.
Interfacing LCD/7-segment display with MCU and putting everything on board.

Proposed:	15th March	20th March	27th March	3rd April	15th April
Achieved:	15th March	20th March	30th March	18rd April	23th April

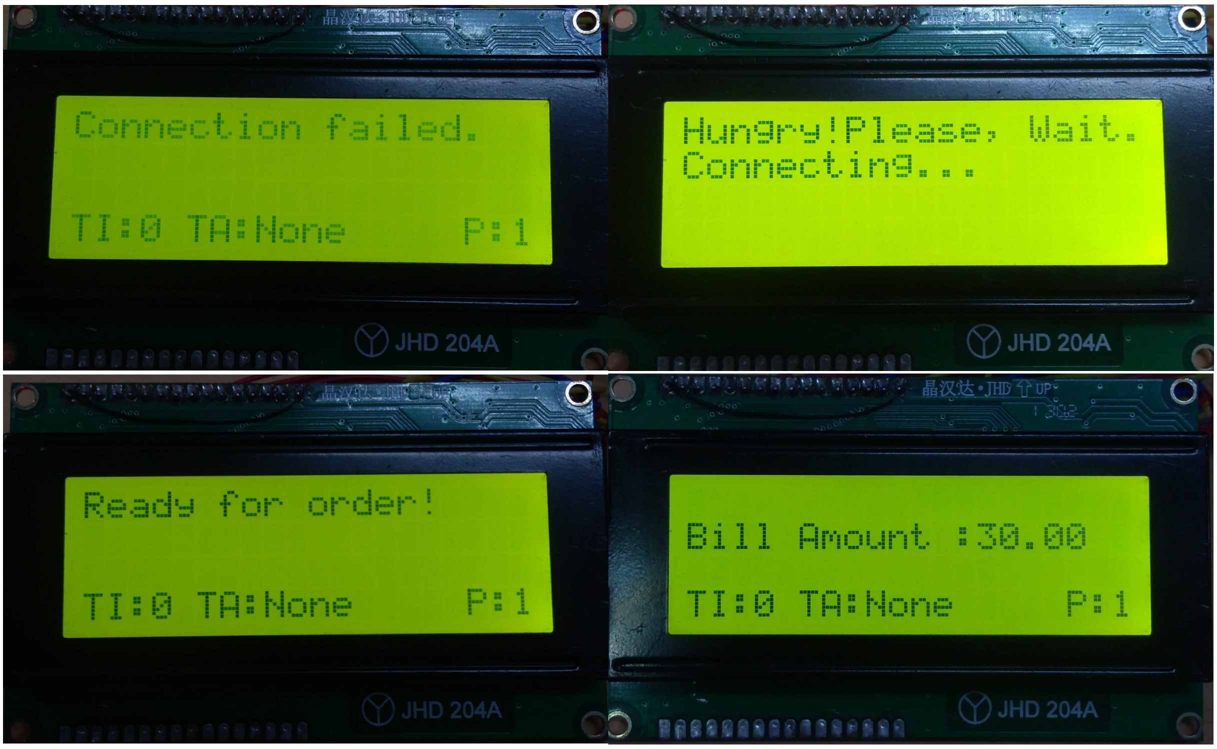
7 Results

7.1 Overall Final Layout and Architecture

Figure 3: Final External View of Project



7.2 User Input and Server Response on LCD



```
Connection failed.

TI:0 TA:None      P:1
```

```
Hungry!Please, Wait.
Connecting...
```

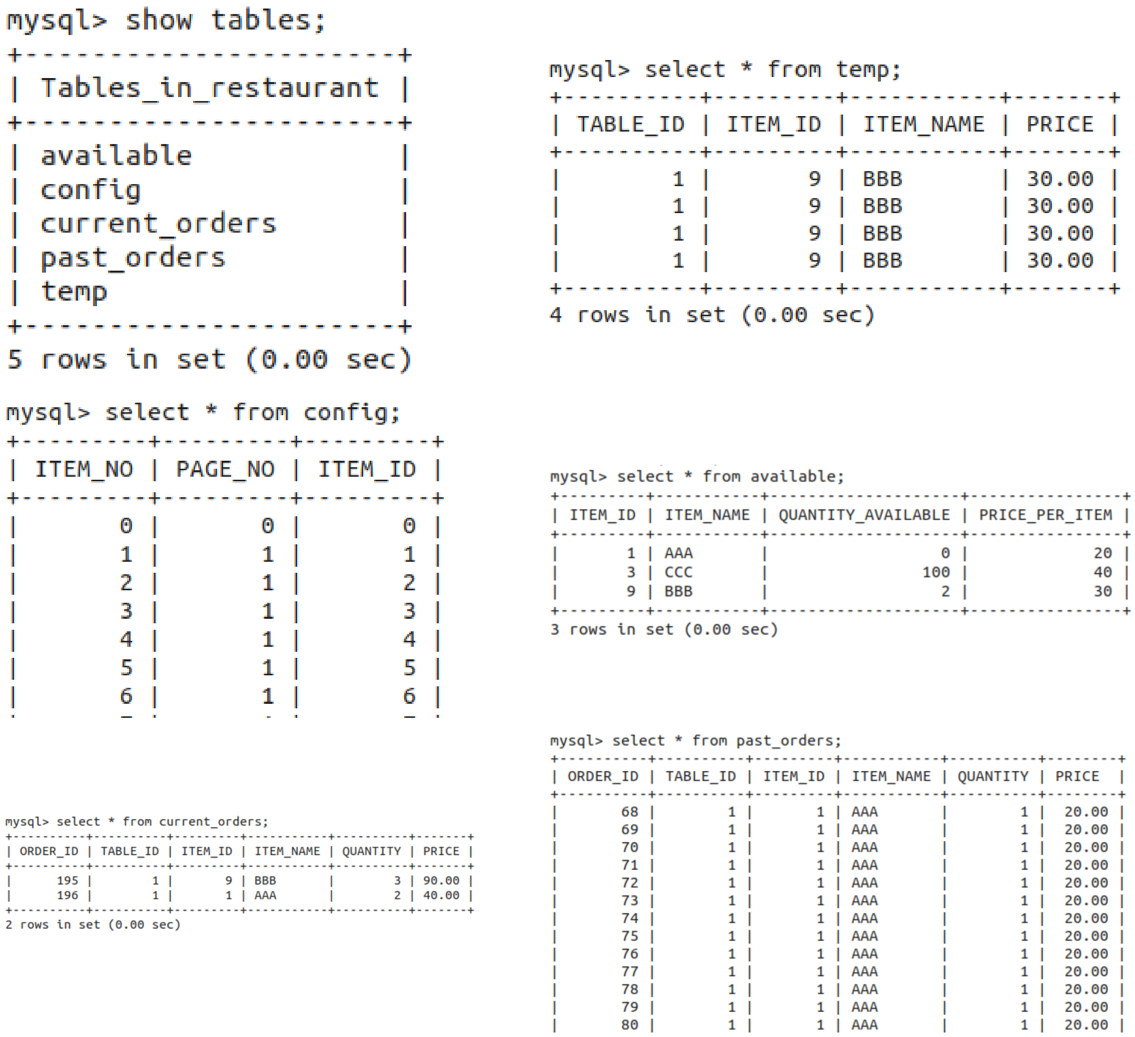
```
Ready for order!

TI:0 TA:None      P:1
```

```
Bill Amount :30.00
TI:0 TA:None      P:1
```

Figure 4: LCD Output

7.3 MySQL Database Tables and Config File



```
mysql> show tables;
+-----------------------+
| Tables_in_restaurant |
+-----------------------+
| available             |
| config                |
| current_orders        |
| past_orders           |
| temp                  |
+-----------------------+
5 rows in set (0.00 sec)
```

```
mysql> select * from config;
+---------+---------+---------+
| ITEM_NO | PAGE_NO | ITEM_ID |
+---------+---------+---------+
|       0 |       0 |       0 |
|       1 |       1 |       1 |
|       2 |       1 |       2 |
|       3 |       1 |       3 |
|       4 |       1 |       4 |
|       5 |       1 |       5 |
|       6 |       1 |       6 |
```

```
mysql> select * from current_orders;
+----------+----------+---------+-----------+----------+-------+
| ORDER_ID | TABLE_ID | ITEM_ID | ITEM_NAME | QUANTITY | PRICE |
+----------+----------+---------+-----------+----------+-------+
|      195 |        1 |       9 | BBB       |        3 | 90.00 |
|      196 |        1 |       1 | AAA       |        2 | 40.00 |
+----------+----------+---------+-----------+----------+-------+
2 rows in set (0.00 sec)
```

```
mysql> select * from temp;
+----------+---------+-----------+-------+
| TABLE_ID | ITEM_ID | ITEM_NAME | PRICE |
+----------+---------+-----------+-------+
|        1 |       9 | BBB       | 30.00 |
|        1 |       9 | BBB       | 30.00 |
|        1 |       9 | BBB       | 30.00 |
|        1 |       9 | BBB       | 30.00 |
+----------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

```
mysql> select * from available;
+---------+-----------+--------------------+----------------+
| ITEM_ID | ITEM_NAME | QUANTITY_AVAILABLE | PRICE_PER_ITEM |
+---------+-----------+--------------------+----------------+
|       1 | AAA       |                  0 |             20 |
|       3 | CCC       |                100 |             40 |
|       9 | BBB       |                  2 |             30 |
+---------+-----------+--------------------+----------------+
3 rows in set (0.00 sec)
```

```
mysql> select * from past_orders;
+----------+----------+---------+-----------+----------+--------+
| ORDER_ID | TABLE_ID | ITEM_ID | ITEM_NAME | QUANTITY | PRICE  |
+----------+----------+---------+-----------+----------+--------+
|       68 |        1 |       1 | AAA       |        1 |  20.00 |
|       69 |        1 |       1 | AAA       |        1 |  20.00 |
|       70 |        1 |       1 | AAA       |        1 |  20.00 |
|       71 |        1 |       1 | AAA       |        1 |  20.00 |
|       72 |        1 |       1 | AAA       |        1 |  20.00 |
|       73 |        1 |       1 | AAA       |        1 |  20.00 |
|       74 |        1 |       1 | AAA       |        1 |  20.00 |
|       75 |        1 |       1 | AAA       |        1 |  20.00 |
|       76 |        1 |       1 | AAA       |        1 |  20.00 |
|       77 |        1 |       1 | AAA       |        1 |  20.00 |
|       78 |        1 |       1 | AAA       |        1 |  20.00 |
|       79 |        1 |       1 | AAA       |        1 |  20.00 |
|       80 |        1 |       1 | AAA       |        1 |  20.00 |
```

Figure 5: MySQL Database Tables Created

References

[1] *7-Segment Display Interfacing with Arduino UNO*. https://www.electronicwings.com/arduino/7-segment-display-interfacing-with-arduino-uno.

[2] *ESP8266 module comparison*. https://blog.squix.org/2015/03/esp8266-module-comparison-esp-01-esp-05.html.

[3] *Getting Started with ESP8266 and Arduino: ESP8266 Arduino Interface*. https://www.electronicshub.org/esp8266-arduino-interface/.

[4] *How to Connect ESP8266 to WiFi | A Beginner's Guide*. https://www.electronicshub.org/connect-esp8266-to-wifi/.

[5] *How to Send Data from Arduino to Webpage using WiFi*. https://circuitdigest.com/microcontroller-projects/sending-arduino-data-to-webpage.

[6] *Interfacing 16X2 LCD with ESP32 using I2C*. https://circuitdigest.com/microcontroller-projects/interfacing-16x2-lcd-with-esp32-using-i2c.

[7] *Serially Interfaced, 8-Digit, LED Display Drivers*. https://www.maximintegrated.com/en/products/power/display-power-control/MAX7221.html.