<u>DATABASE MANAGEMENT SYSTEM</u>

HOTEL MANAGEMENT

By:

Konda Baghirath Reddy
[19BCE2526]

Srushti Suresh Jagtap
[19BCE0325]

Harshit Kumar
[19BCE2113]

<u>CONTENTS</u>

ABSTARCT

This project aims to develop a computerized system for seamless booking and management of rooms and maintaining bills of the customers in the hotels. This project is equipped with menu options to perform various managerial tasks regarding everything from room booking to bills management. Most hotel management application are very limited in terms of functionality and applicability and the hotel management tasks has to be done across various other software, but our project aims to deliver a software that can be used across all domains of hotel management business. This application helps us in viewing all the rooms booked in the hotels under the owner. These records can’t be deleted from inside the software making it tamper-proof too. Overall, this project of ours is being developed to help hoteliers manage their hotels in a seamless manner.

INTRODUCTION

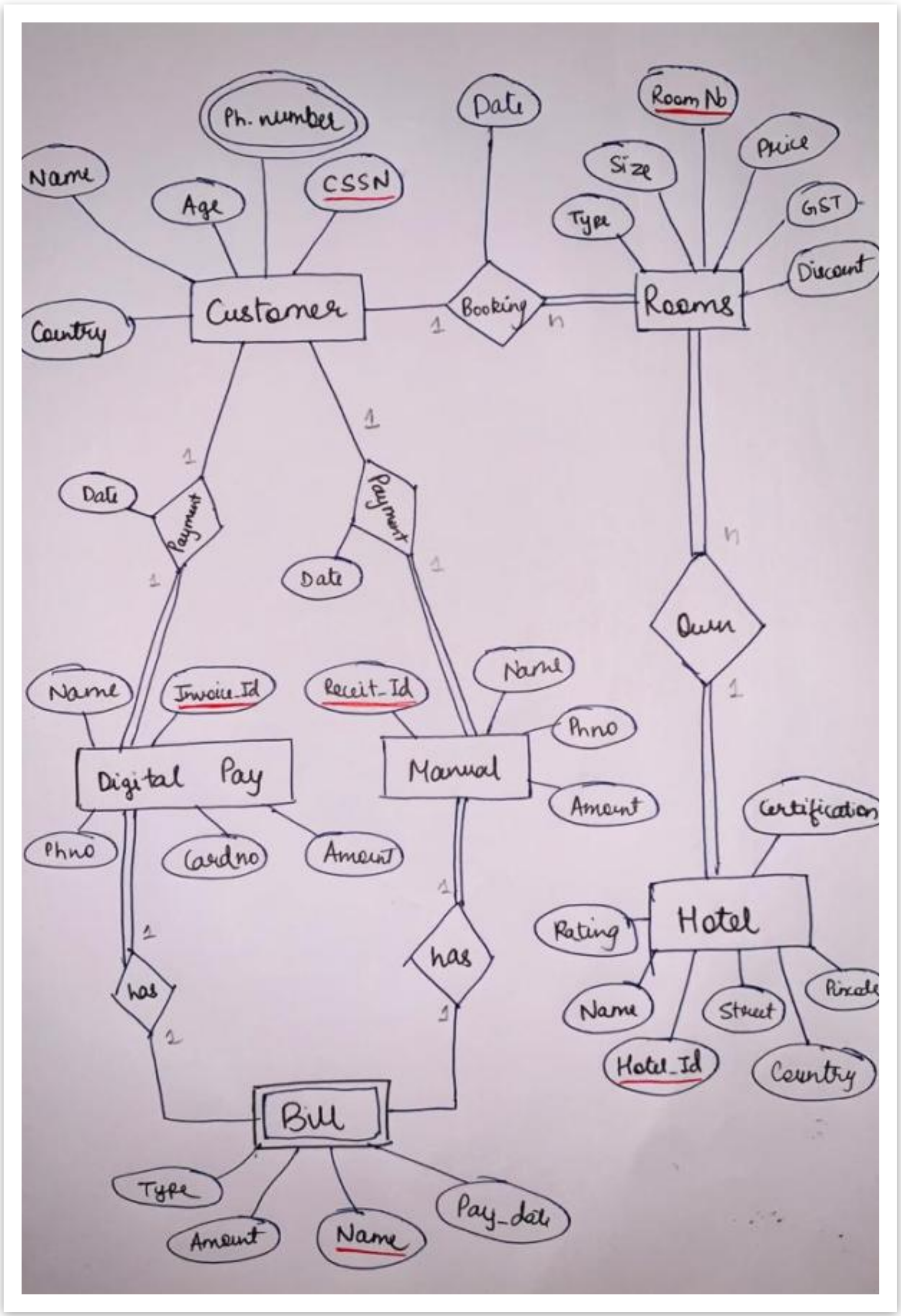
The project, Hotel management System is a mobile- based application that allows the manager to handle all hotel activities. This application gives him the power to manage the entire system. Interactive interface and the ability to manage various hotel booking and rooms make this system very flexible and convenient. Hotel management project allow manager to view the bills, room booking, type of payment and other necessary management features.

The project has a wide scope, which can be applied by any business organizations. This project is designed for employers. It has room booking service which can keep track of reservations and room availability and also have tracking of bills which can help finance department. The service can be used by hotel manager and owner to access any room booking system and bill tracking system.

TECHNOLOGIES USED:

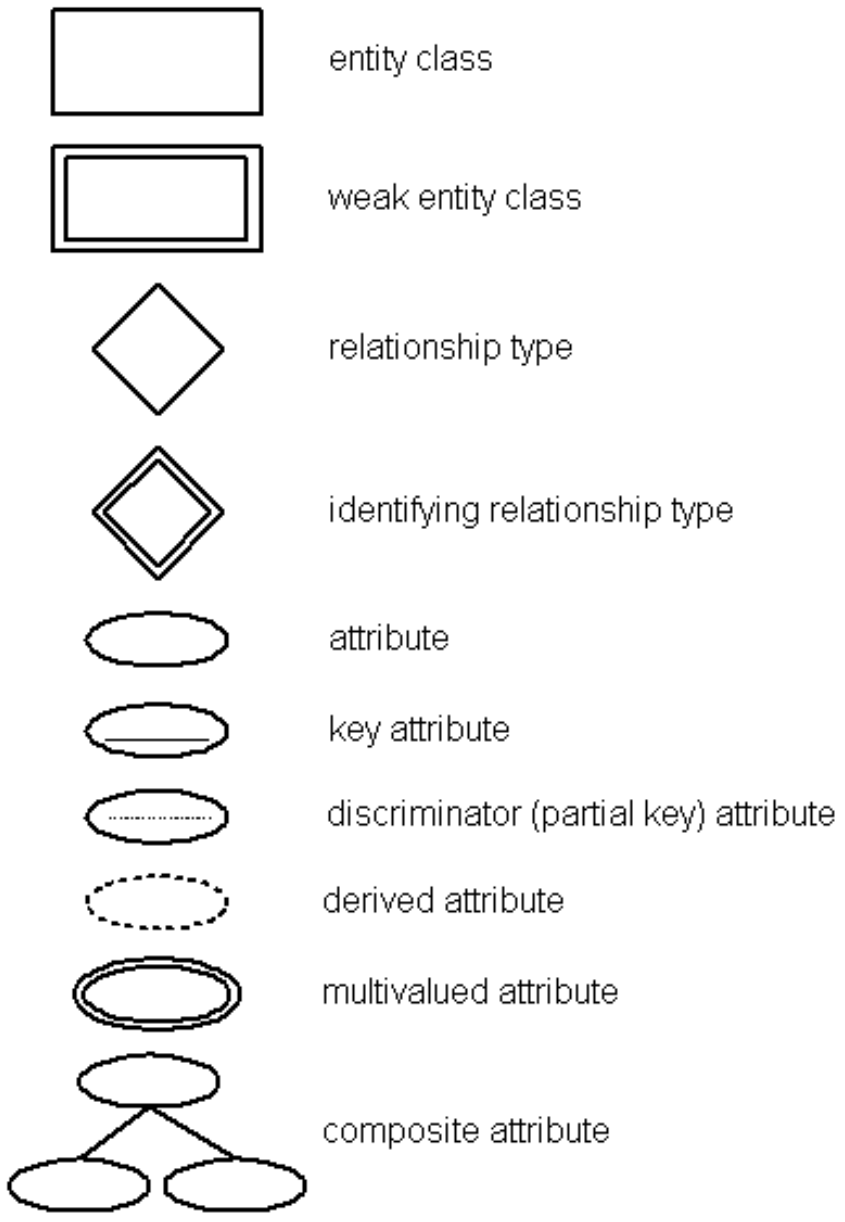
- FLUTTER
- NODE JS
- MONGO DB

ER DIAGRAM



Ph. number
Date
Room No
Name
Age
CSSN
Size
Price
Type
GST
Discount
Customer
Booking
Rooms
Country
1
n
1
1
Date
Payment
Payment
Date
1
1
n
Own
1
Name
Invoice_Id
Receit_Id
Name
Phno
Digital Pay
Manual
Amount
Certification
Phno
Cardno
Amount
1
Hotel
Rating
1
has
has
Pincode
Name
Street
1
1
Hotel_Id
Country
Bill
Type
Pay_date
Amount
Name

Notations used in ER diagram:



Conclusion:

Strong Entities:

In the ER diagram hotel, customer, rooms, digital payment, bill, Location. They all act as strong entities since they have respective unique attributes known as keys, by which we could identify them.

Weak Entities:

Manual and Price are are the weak entities in ER diagram. Since they don't have a definite key they only have a partial key.

Primary keys:

customer-(cssn); digital- (invoice.id); bill-(receipt.id); hotel-(hotel.id); rooms-(room.no); Location-(address)

Foreign keys:

price-(type/size); Invoice-(email/ph-num); Manual-(name/ph.num); Bill-(invoice.id/recipt.id)

Relationships:

Customer->payment->digital Customer->payment->manual Customer->booking->rooms Digital->has->bill Manual->has->bill Hotel->owns->rooms Price ->of->Rooms

ER DIAGRAM TO SCHEMA CONVERSION

1.**CUSTOMER** (**CSSN**, NAME, AGE, PHNO, COUNTRY)

2.**MANUAL** (AMOUNT, PHNO, NAME, CSSN, DATE, **RECEIPTID**) - Foreign key=(CSSN)

3.**DIGITAL** PAY (NAME, PHNO, CARDNO, AMOUNT, DATE, CSSN, **INVOICEID**) – Foreign key=(CSSN)

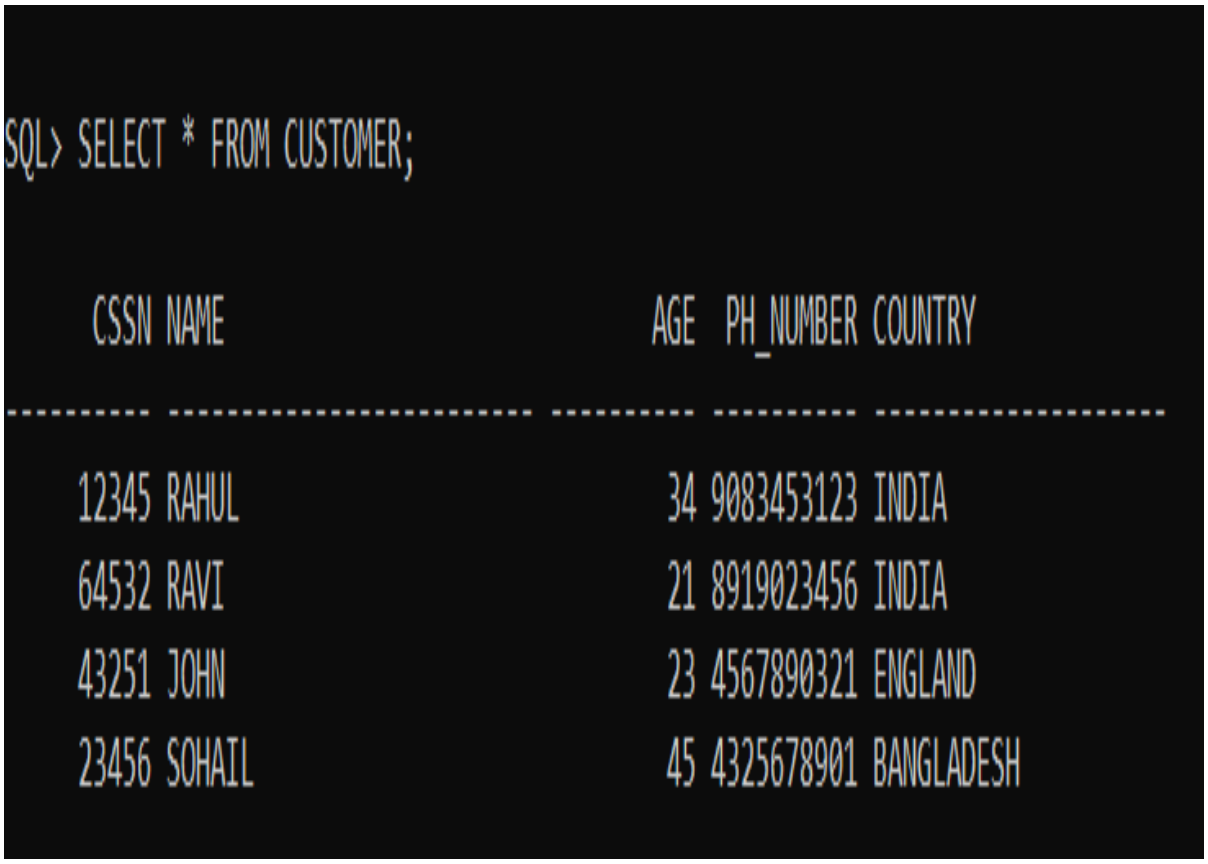
4.**BILL** (AMOUNT, **NAME**, TYPE, PAYDATE, **RECEIPTID**, **INVOICEID**) – Foreign key = (RECEIPTID, INVOICEID)

5.**ROOM** (TYPE, SIZE, DISCOUNT, GST, PRICE, SDATE, EDATE, HOTELID, CSSN, **ROOMNO**) Foreign keys=(CSSN, HOTELID), HERE (CHECK_IN_DATE,CHECK_OUT_DATE)=BOOKING_DATE

6.**HOTEL** (NAME, CERTIFICATION, **HOTEL ID**, RATING, COUNTRY, STREET, PINCODE)

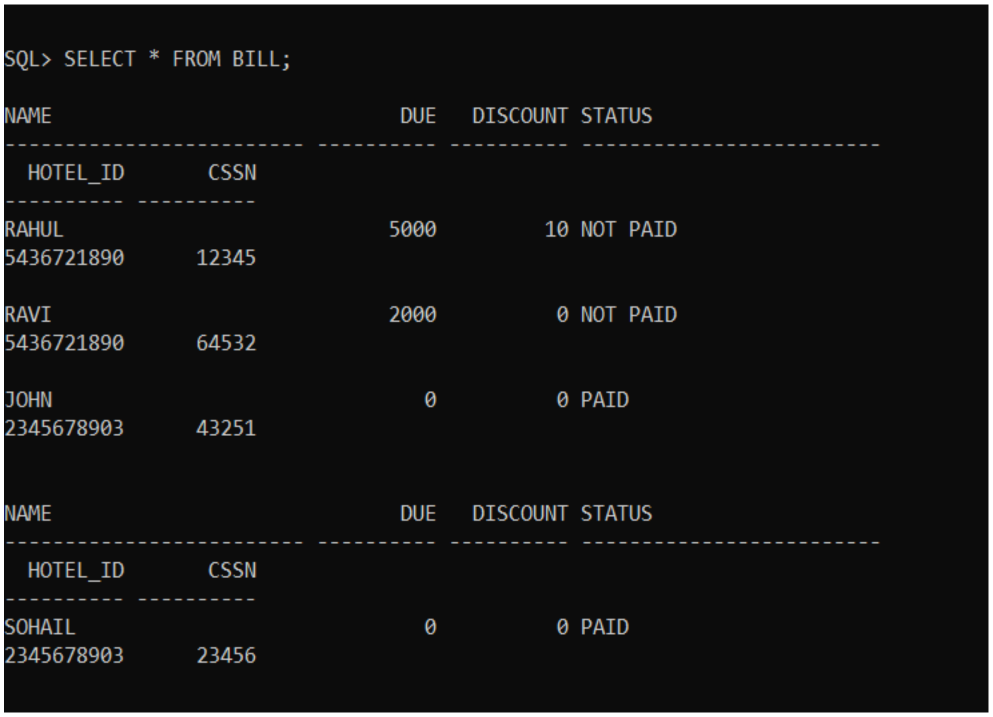
CREATION OF RELATIONAL TABLES

1. Customers:



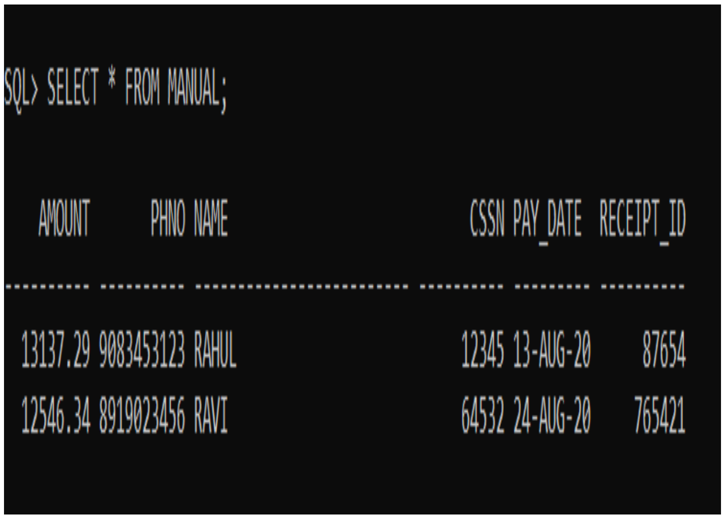
```
SQL> SELECT * FROM CUSTOMER;

     CSSN NAME                           AGE  PH_NUMBER COUNTRY
--------- ---------------------- --------- --------- ------------------
    12345 RAHUL                           34 9083453123 INDIA
    64532 RAVI                            21 8919023456 INDIA
    43251 JOHN                            23 4567890321 ENGLAND
    23456 SOHAIL                          45 4325678901 BANGLADESH
```

2. Bills:



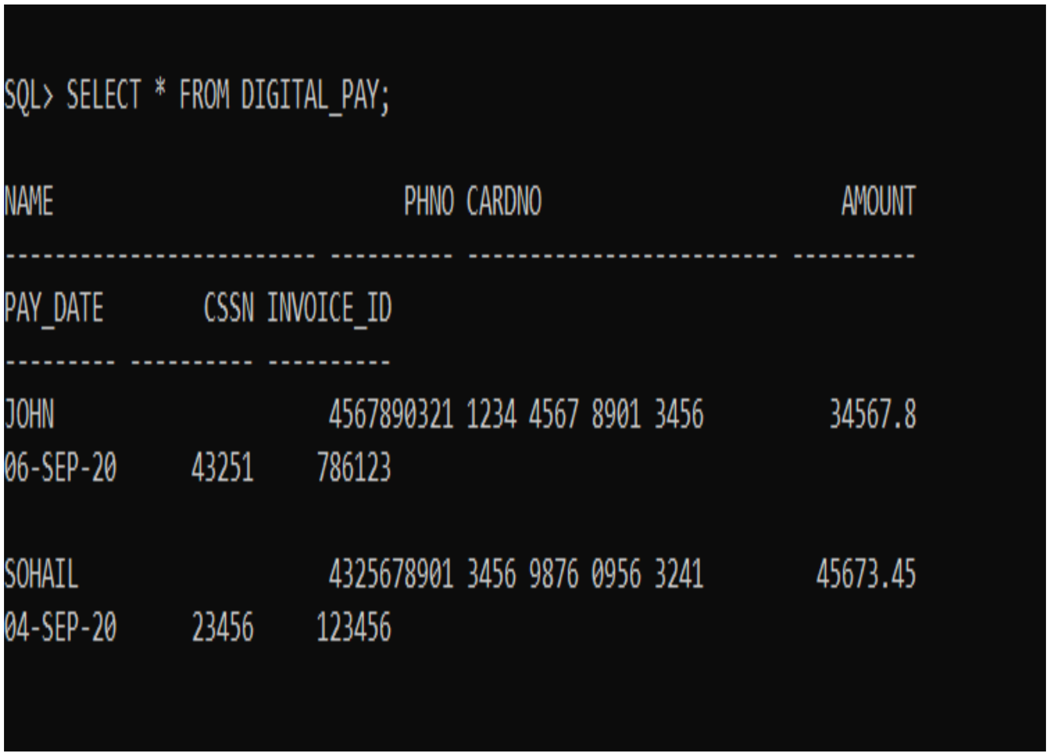
```
SQL> SELECT * FROM BILL;

NAME                             DUE   DISCOUNT STATUS
------------------------- ---------- ---------- -------------------------
  HOTEL_ID       CSSN
---------- ----------
RAHUL                           5000         10 NOT PAID
5436721890      12345

RAVI                            2000          0 NOT PAID
5436721890      64532

JOHN                               0          0 PAID
2345678903      43251


NAME                             DUE   DISCOUNT STATUS
------------------------- ---------- ---------- -------------------------
  HOTEL_ID       CSSN
---------- ----------
SOHAIL                             0          0 PAID
2345678903      23456
```

3. Manual Payments



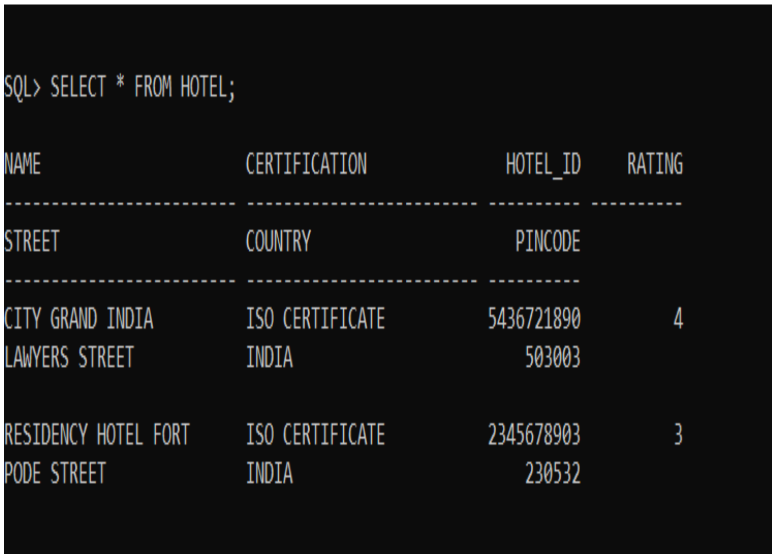
```
SQL> SELECT * FROM MANUAL;

    AMOUNT       PHNO NAME                        CSSN PAY_DATE  RECEIPT_ID
---------- ---------- ------------------------- ---------- --------- ----------
  13137.29 9083453123 RAHUL                      12345 13-AUG-20      87654
  12546.34 8919023456 RAVI                       64532 24-AUG-20     765421
```

4. Digital Payments:



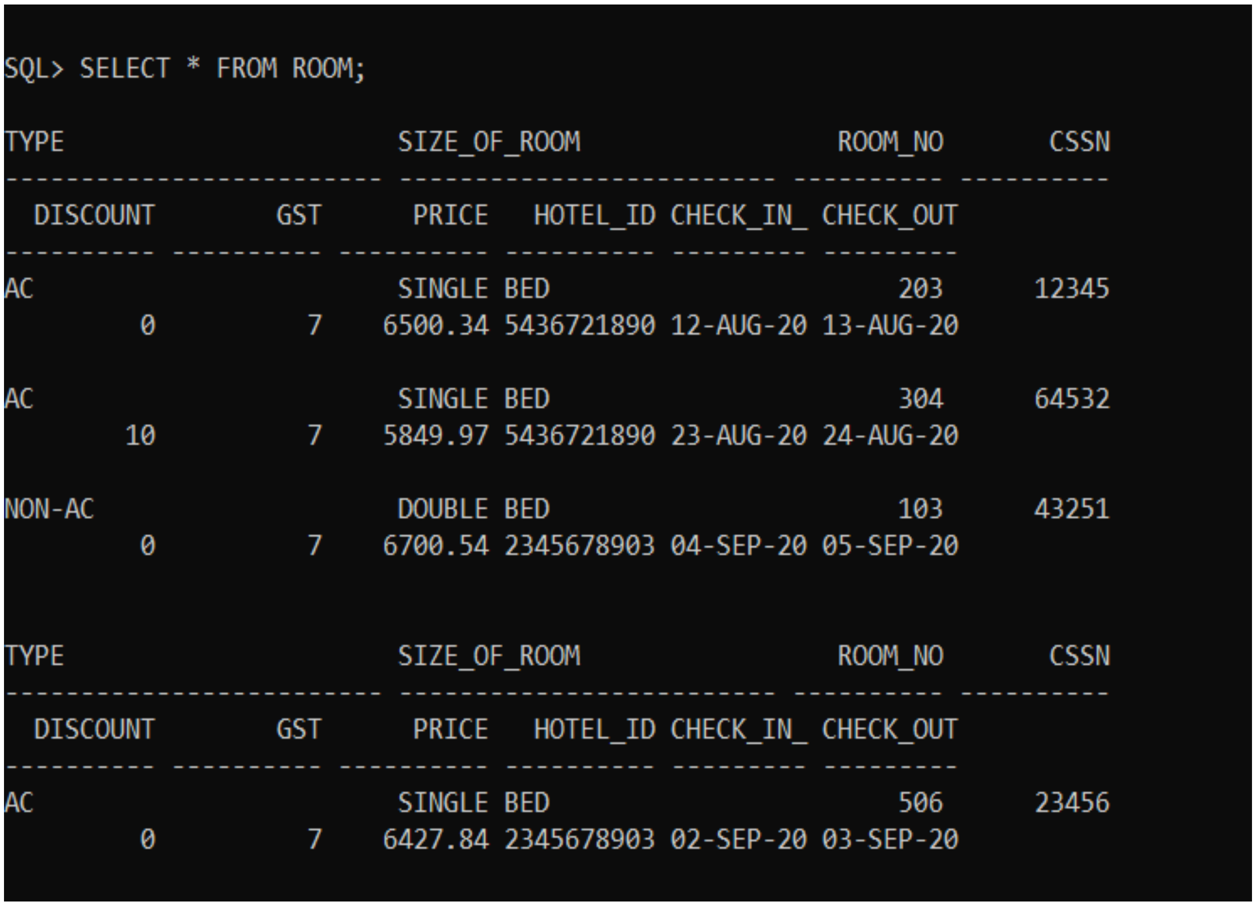
```
SQL> SELECT * FROM DIGITAL_PAY;

NAME                            PHNO CARDNO                      AMOUNT
------------------------- ---------- ------------------------- ----------
PAY_DATE        CSSN INVOICE_ID
--------- ---------- ----------
JOHN                      4567890321 1234 4567 8901 3456          34567.8
06-SEP-20      43251     786123

SOHAIL                    4325678901 3456 9876 0956 3241         45673.45
04-SEP-20      23456     123456
```

5. Hotels:



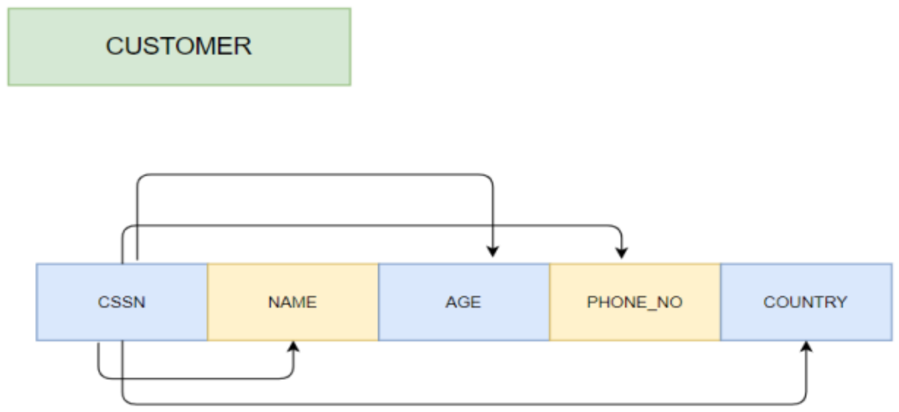
```
SQL> SELECT * FROM HOTEL;

NAME                     CERTIFICATION              HOTEL_ID     RATING
------------------------ ------------------------ ---------- ----------
STREET                   COUNTRY                    PINCODE
------------------------ ------------------------ ----------
CITY GRAND INDIA         ISO CERTIFICATE          5436721890          4
LAWYERS STREET           INDIA                        503003

RESIDENCY HOTEL FORT     ISO CERTIFICATE          2345678903          3
PODE STREET              INDIA                        230532
```

6. Rooms:



```
SQL> SELECT * FROM ROOM;

TYPE                      SIZE_OF_ROOM                 ROOM_NO       CSSN
------------------------- ------------------------- ---------- ----------
  DISCOUNT        GST      PRICE   HOTEL_ID CHECK_IN_ CHECK_OUT
---------- ---------- ---------- ---------- --------- ---------
AC                        SINGLE BED                       203      12345
         0          7    6500.34 5436721890 12-AUG-20 13-AUG-20

AC                        SINGLE BED                       304      64532
        10          7    5849.97 5436721890 23-AUG-20 24-AUG-20

NON-AC                    DOUBLE BED                       103      43251
         0          7    6700.54 2345678903 04-SEP-20 05-SEP-20


TYPE                      SIZE_OF_ROOM                 ROOM_NO       CSSN
------------------------- ------------------------- ---------- ----------
  DISCOUNT        GST      PRICE   HOTEL_ID CHECK_IN_ CHECK_OUT
---------- ---------- ---------- ---------- --------- ---------
AC                        SINGLE BED                       506      23456
         0          7    6427.84 2345678903 02-SEP-20 03-SEP-20
```

NORMALIZATION OF THE TABLE



Functional Dependencies:

CSSN->NAME, AGE, PHONE_NO, COUNTRY

PHONE_NO -> CSSN, NAME, AGE, COUNTRY (since here phone_no is unique)

CSSN$^+$ ={CSSN, NAME, AGE, PHONE_NO, COUNTRY }

PHONE_NO$^+$ ={CSSN, NAME, AGE, PHONE_NO, COUNTRY }

Candidate keys={CSSN, PHONE_NO}

PRIMARY ATTRIBUTES = { CSSN, PHONE_NO}

NON-PRIMARY ATTRIBUTES = { NAME, AGE, COUNTRY}

1NF:

Here all the attributes are single valued. So, it is in 1NF.

2NF:

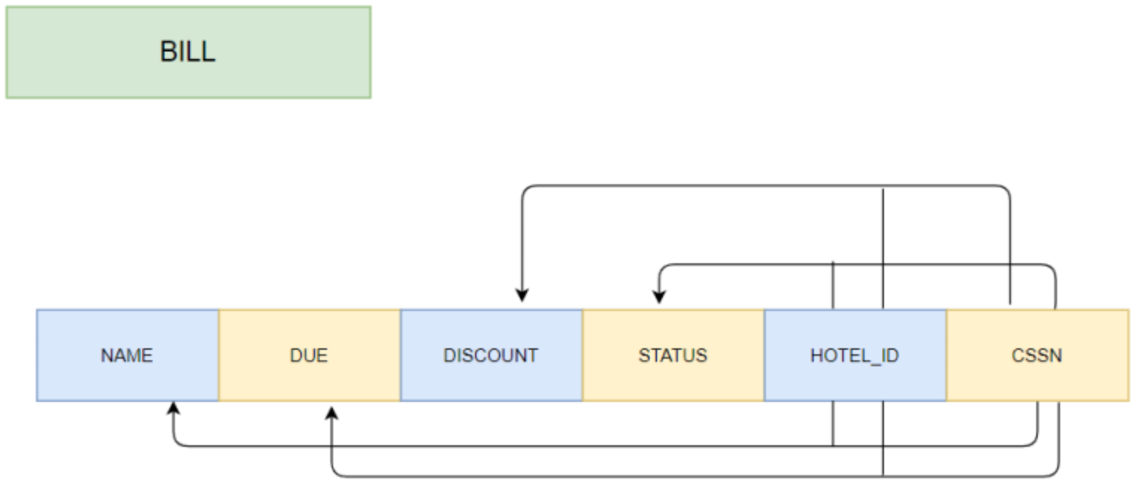
It is in 1NF and there are no partial dependency. So, it is in 2NF.

3NF:
It is in 1NF, 2NF and there are no transitive dependency. So, it is in 3NF

BCNF:

Here the LHS of the FD are candidate keys

So, as the table is in BCNF.



Functional Dependencies:

CSSN HOTEL_ID -> NAME, DUE, DISCOUNT, STATUS

CSSN HOTEL_ID$^+$ ={ NAME DUE DISCOUNT STATUS CSSN HOTEL_ID }

Candidate keys={ CSSN HOTEL_ID }

PRIMARY ATTRIBUTES = { CSSN, HOTEL_ID }

NON-PRIMARY ATTRIBUTES = { NAME, DUE, DISCOUNT, STATUS }

1NF:

Here all the attributes are single valued. So, it is in 1NF.

2NF:

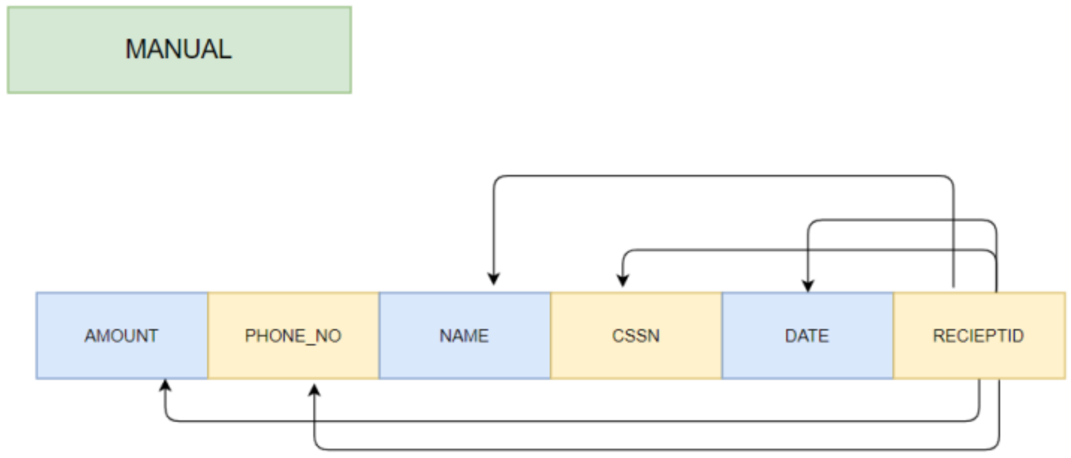
It is in 1NF and there is no partial dependency. So, it is in 2NF.

3NF:
It is in 1NF, 2NF and there is no transitive dependency. So, it is in 3NF

BCNF:

Here the LHS of the FD (CSSN HOTEL_ID) is candidate keys

So, as the table is in BCNF.



Functional Dependencies:

RECIEPT_ID -> AMOUNT, PHONE_NO, NAME, CSSN, DATE

RECIEPT_ID$^+$ ={ AMOUNT PHONE_NO NAME CSSN DATE RECIEPT_ID }

Candidate keys={ RECIEPT_ID }

PRIMARY ATTRIBUTES = { RECIEPT_ID }

NON-PRIMARY ATTRIBUTES = { AMOUNT PHONE_NO NAME CSSN DATE }

1NF:

Here all the attributes are single valued. So, it is in 1NF.

2NF:

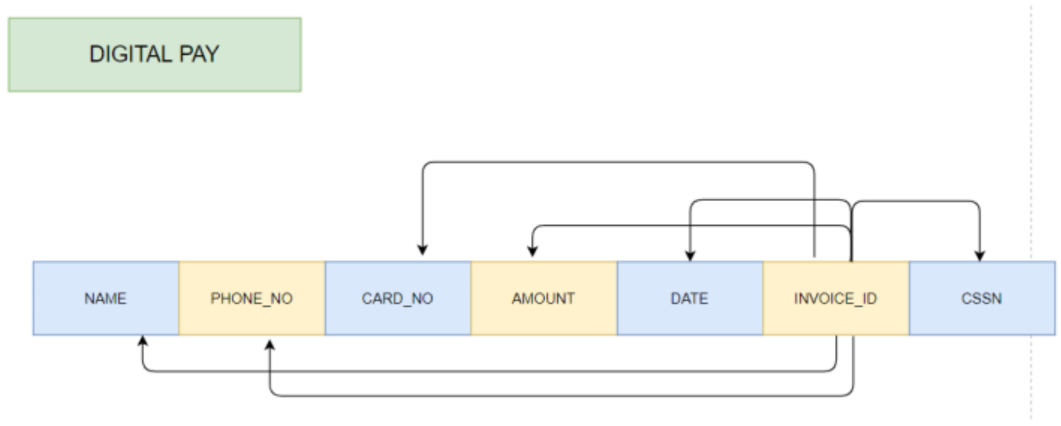
It is in 1NF and there is no partial dependency. So, it is in 2NF.

3NF:
It is in 1NF, 2NF and there is no transitive dependency. So, it is in 3NF

BCNF:

Here the LHS of the FD RECIEPT_ID is candidate keys

So, as the table is in BCNF.



Functional Dependencies:

INVOICE_ID -> AMOUNT, PHONE_NO, NAME, CSSN, DATE, CARD_NO

INVOICE_ID^{+} ={ INVOICE_ID AMOUNT PHONE_NO NAME CSSN DATE CARD_NO

}

Candidate keys={INVOICE_ID }

PRIMARY ATTRIBUTES = { INVOICE_ID }

NON-PRIMARY ATTRIBUTES = { AMOUNT PHONE_NO NAME CSSN DATE CARD_NO }

1NF:

Here all the attributes are single valued. So, it is in 1NF.

2NF:

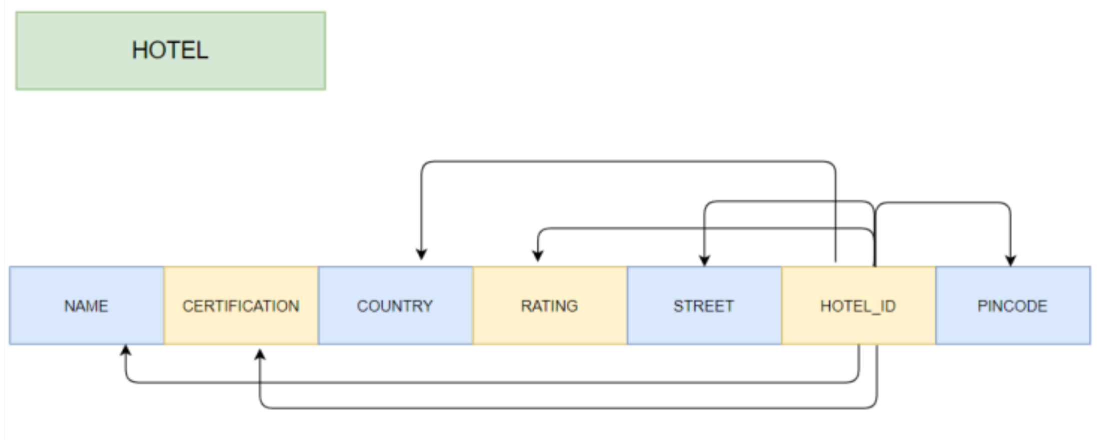
It is in 1NF and there is no partial dependency. So, it is in 2NF.

3NF:
It is in 1NF, 2NF and there is no transitive dependency. So, it is in 3NF

BCNF:

Here the LHS of the FD INVOICE_ID is candidate keys

So, as the table is in BCNF.



Functional Dependencies:

HOTEL_ID -> NAME, CERTIFICATION, COUNTRY, RATING, STREET, PINCODE

HOTEL_ID$^+$ ={NAME CERTIFICATION COUNTRY RATING STREET HOTEL_ID PINCODE }

Candidate keys={ HOTEL_ID }

PRIMARY ATTRIBUTES = { HOTEL_ID }

NON-PRIMARY ATTRIBUTES = { NAME CERTIFICATION COUNTRY RATING STREET PINCODE }

1NF:

Here all the attributes are single valued. So, it is in 1NF.

2NF:

It is in 1NF and there is no partial dependency. So, it is in 2NF.

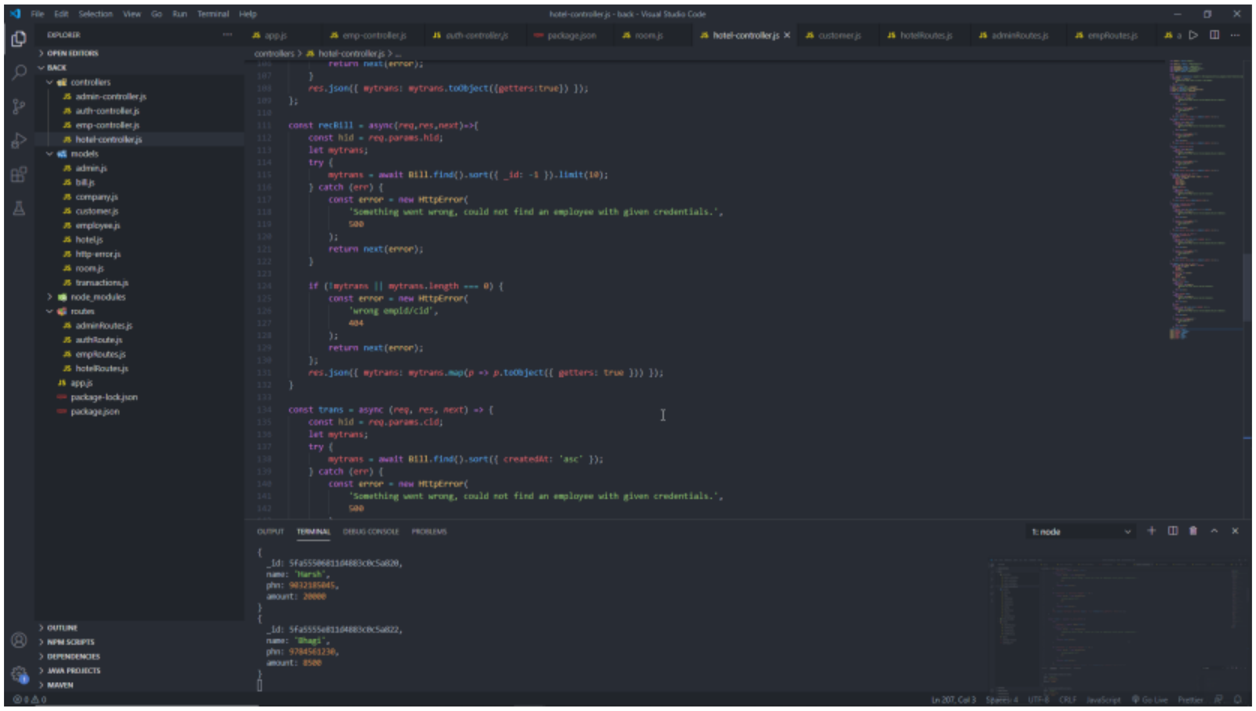
3NF:
It is in 1NF, 2NF and there is no transitive dependency. So, it is in 3NF

BCNF:

Here the LHS of the FD HOTEL_ID is candidate keys

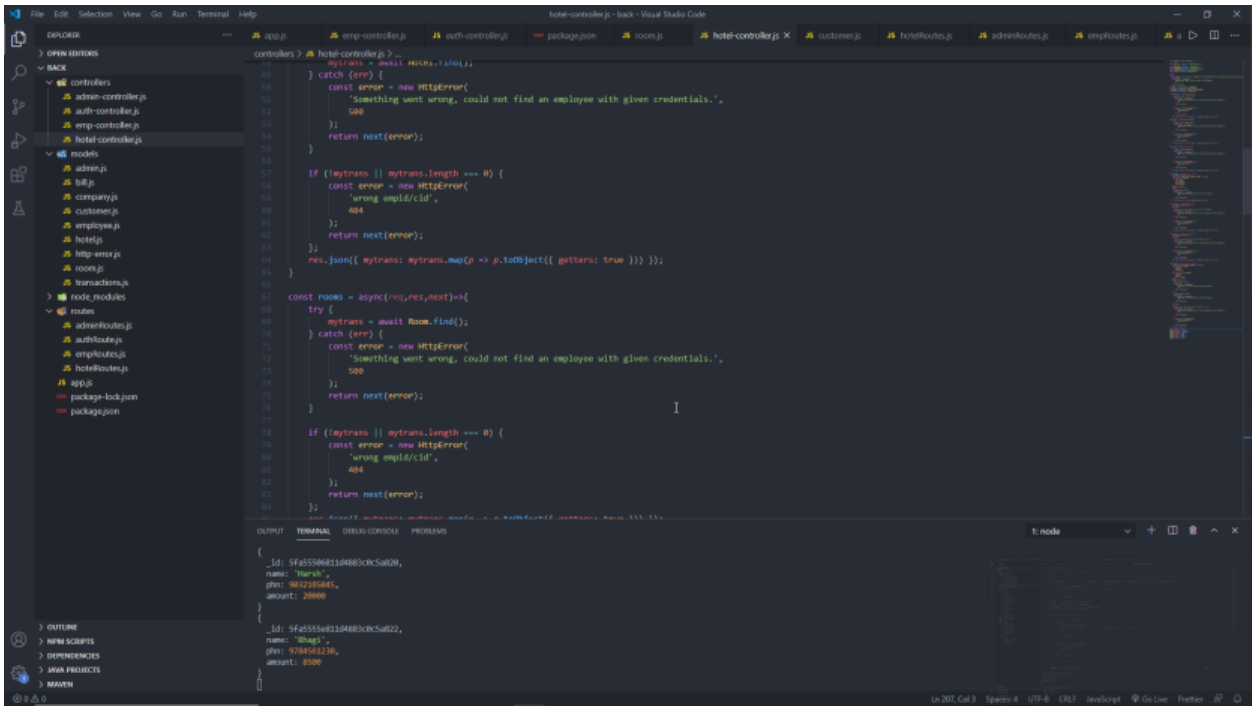
So, as the table is in BCNF.

CODE FOR DATABSE AND FRONTEND CONNECTIVITY



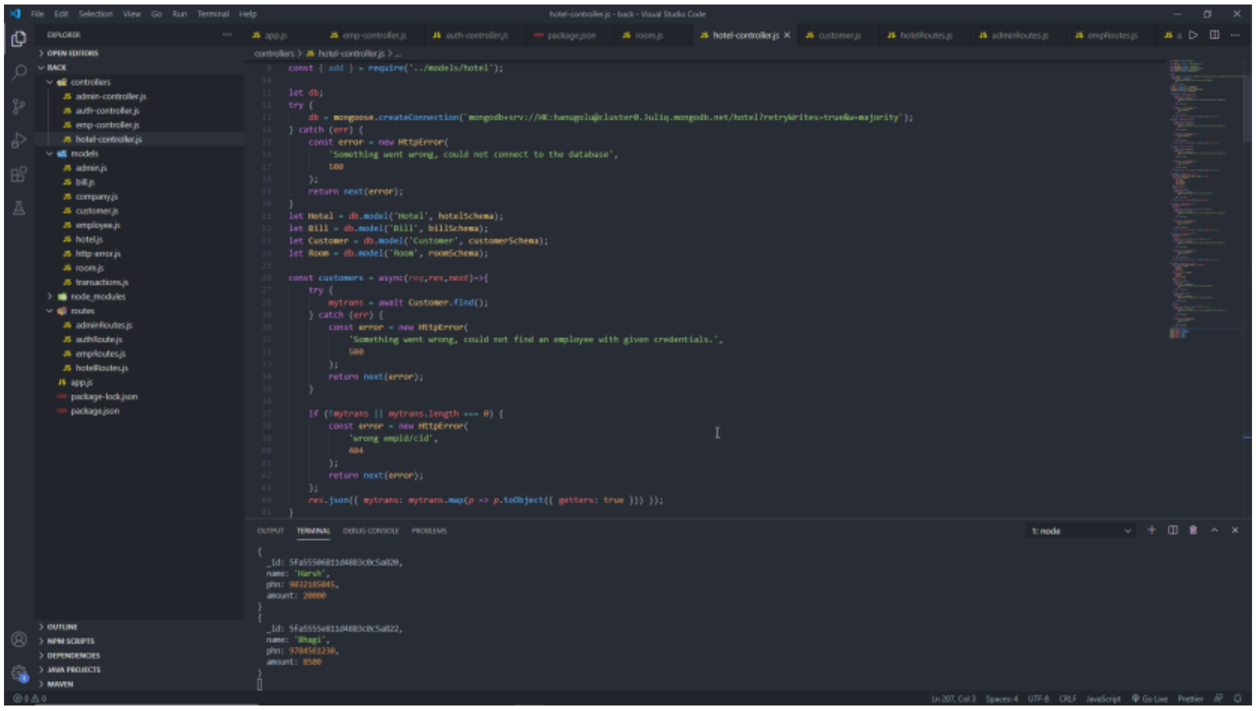
```
            return next(error);
        }
        res.json({ mytrans: mytrans.toObject({getters:true}) });
};

const recBill = async(req,res,next)=>{
    const hid = req.params.hid;
    let mytrans;
    try {
        mytrans = await Bill.find().sort({ _id: -1 }).limit(10);
    } catch (err) {
        const error = new HttpError(
            'Something went wrong, could not find an employee with given credentials.',
            500
        );
        return next(error);
    }

    if (!mytrans || mytrans.length === 0) {
        const error = new HttpError(
            'wrong empId/cid',
            404
        );
        return next(error);
    };
    res.json({ mytrans: mytrans.map(p => p.toObject({ getters: true })) });
}

const trans = async (req, res, next) => {
    const hid = req.params.cid;
    let mytrans;
    try {
        mytrans = await Bill.find().sort({ createdAt: 'asc' });
    } catch (err) {
        const error = new HttpError(
            'Something went wrong, could not find an employee with given credentials.',
            500
```

```
{
  _id: 5fa55506811d4883c0c5a820,
  name: 'Harsh',
  phn: 9032185045,
  amount: 20000
}
{
  _id: 5fa5555e811d4883c0c5a822,
  name: 'Bhagi',
  phn: 9784561230,
  amount: 8500
}
```



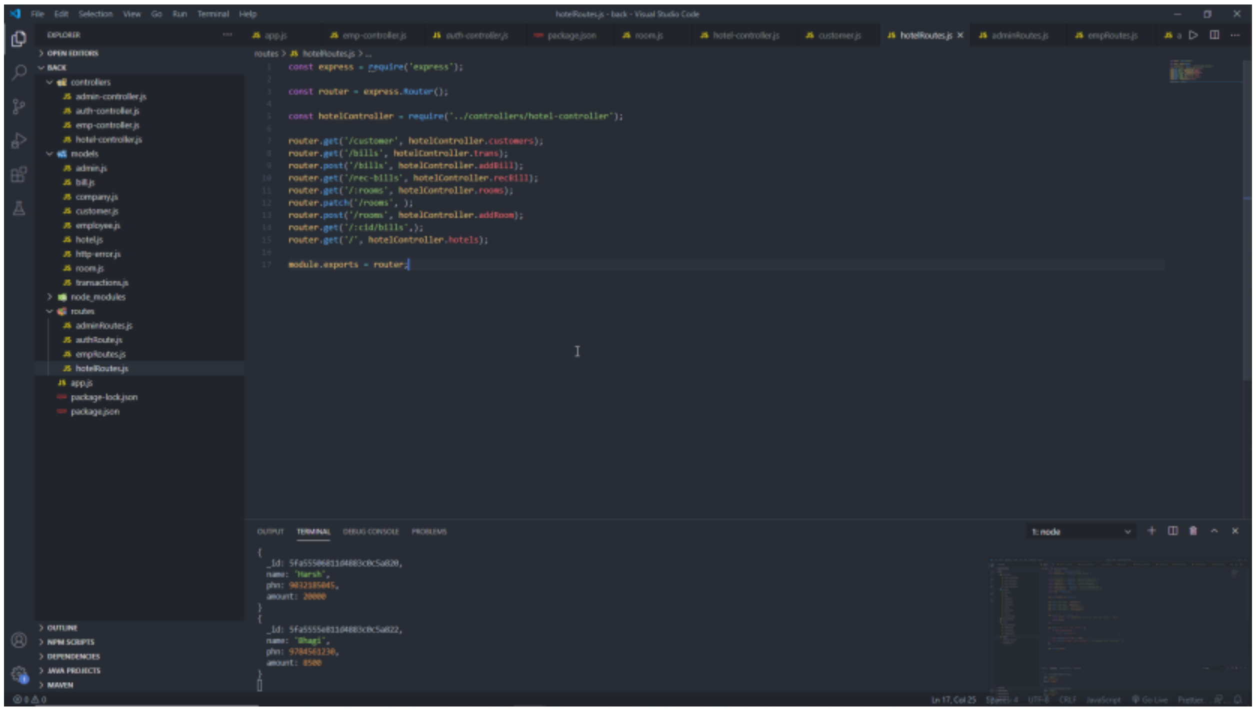
```
        mytrans = await Hotel.find();
    } catch (err) {
        const error = new HttpError(
            'Something went wrong, could not find an employee with given credentials.',
            500
        );
        return next(error);
    }

    if (!mytrans || mytrans.length === 0) {
        const error = new HttpError(
            'wrong empId/cid',
            404
        );
        return next(error);
    };
    res.json({ mytrans: mytrans.map(p => p.toObject({ getters: true })) });
}

const rooms = async(req,res,next)=>{
    try {
        mytrans = await Room.find();
    } catch (err) {
        const error = new HttpError(
            'Something went wrong, could not find an employee with given credentials.',
            500
        );
        return next(error);
    }

    if (!mytrans || mytrans.length === 0) {
        const error = new HttpError(
            'wrong empId/cid',
            404
        );
        return next(error);
    };
```

```
{
  _id: 5fa55506811d4883c0c5a820,
  name: 'Harsh',
  phn: 9032185045,
  amount: 20000
}
{
  _id: 5fa5555e811d4883c0c5a822,
  name: 'Bhagi',
  phn: 9784561230,
  amount: 8500
}
```



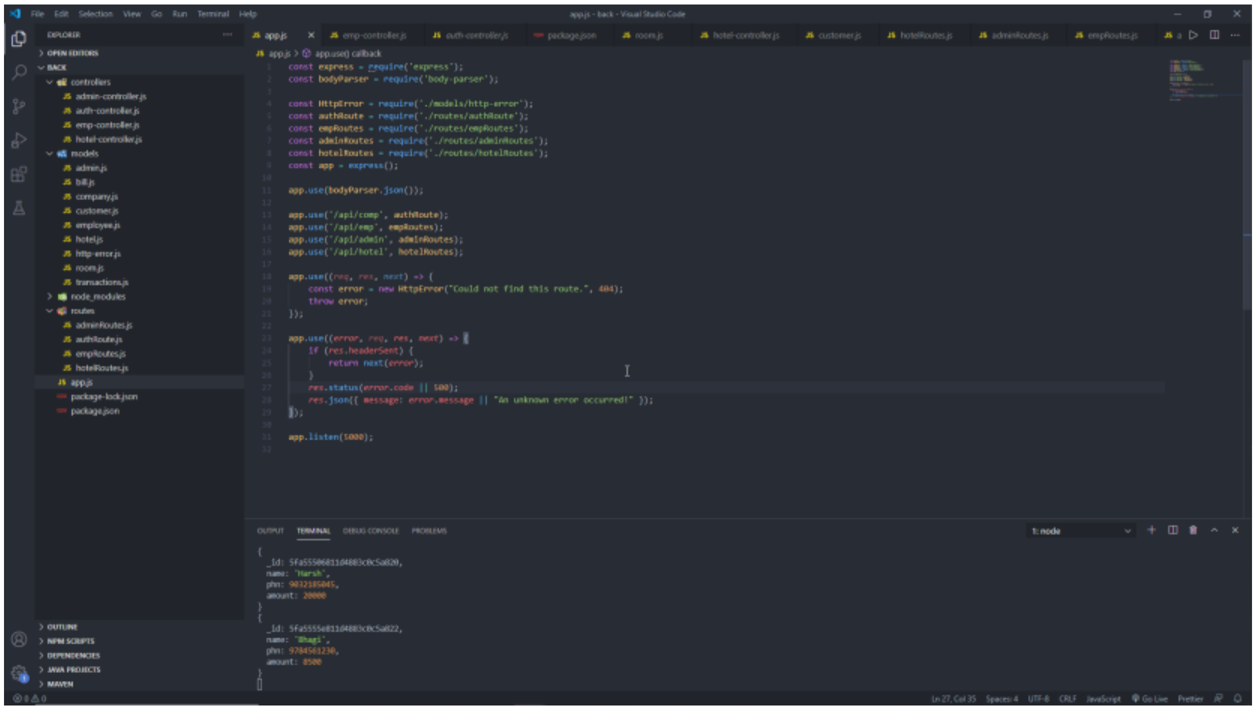
```
const { add } = require('../models/hotel');

let db;
try {
    db = mongoose.createConnection(`mongodb+srv://HK:hanugolu@cluster0.3uliq.mongodb.net/hotel?retryWrites=true&w=majority`);
} catch (err) {
    const error = new HttpError(
        'Something went wrong, could not connect to the database',
        500
    );
    return next(error);
}
let Hotel = db.model('Hotel', hotelSchema);
let Bill = db.model('Bill', billSchema);
let Customer = db.model('Customer', customerSchema);
let Room = db.model('Room', roomSchema);

const customers = async(req,res,next)=>{
    try {
        mytrans = await Customer.find();
    } catch (err) {
        const error = new HttpError(
            'Something went wrong, could not find an employee with given credentials.',
            500
        );
        return next(error);
    }

    if (!mytrans || mytrans.length === 0) {
        const error = new HttpError(
            'wrong empid/cid',
            404
        );
        return next(error);
    };
    res.json({ mytrans: mytrans.map(p => p.toObject({ getters: true })) });
}
```

```
{
  _id: 5fa55506811d4883c0c5a820,
  name: 'Harsh',
  phn: 9032185045,
  amount: 20000
}
{
  _id: 5fa5555e811d4883c0c5a822,
  name: 'Bhagi',
  phn: 9784561230,
  amount: 8500
}
```



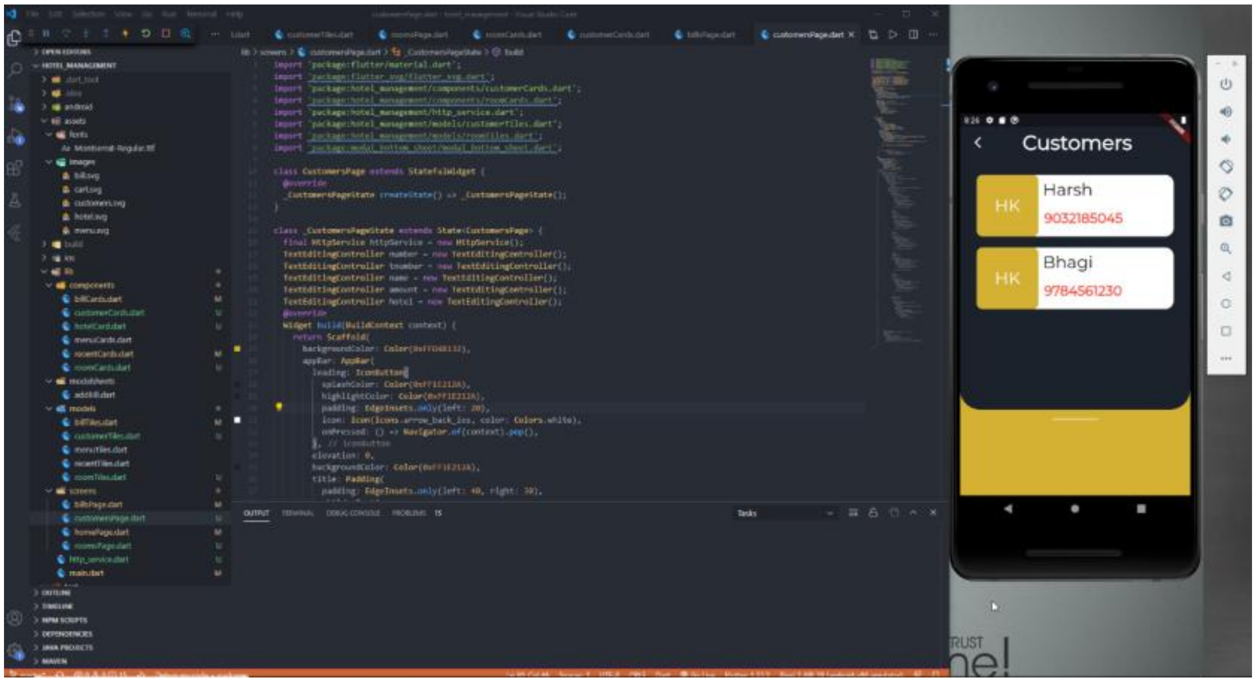
```
const express = require('express');

const router = express.Router();

const hotelController = require('../controllers/hotel-controller');

router.get('/customer', hotelController.customers);
router.get('/bills', hotelController.trans);
router.post('/bills', hotelController.addBill);
router.get('/rec-bills', hotelController.recBill);
router.get('/:rooms', hotelController.rooms);
router.patch('/rooms', );
router.post('/rooms', hotelController.addRoom);
router.get('/:cid/bills',);
router.get('/', hotelController.hotels);

module.exports = router;
```

```
{
  _id: 5fa55506811d4883c0c5a820,
  name: 'Harsh',
  phn: 9032185045,
  amount: 20000
}
{
  _id: 5fa5555e811d4883c0c5a822,
  name: 'Bhagi',
  phn: 9784561230,
  amount: 8500
}
```

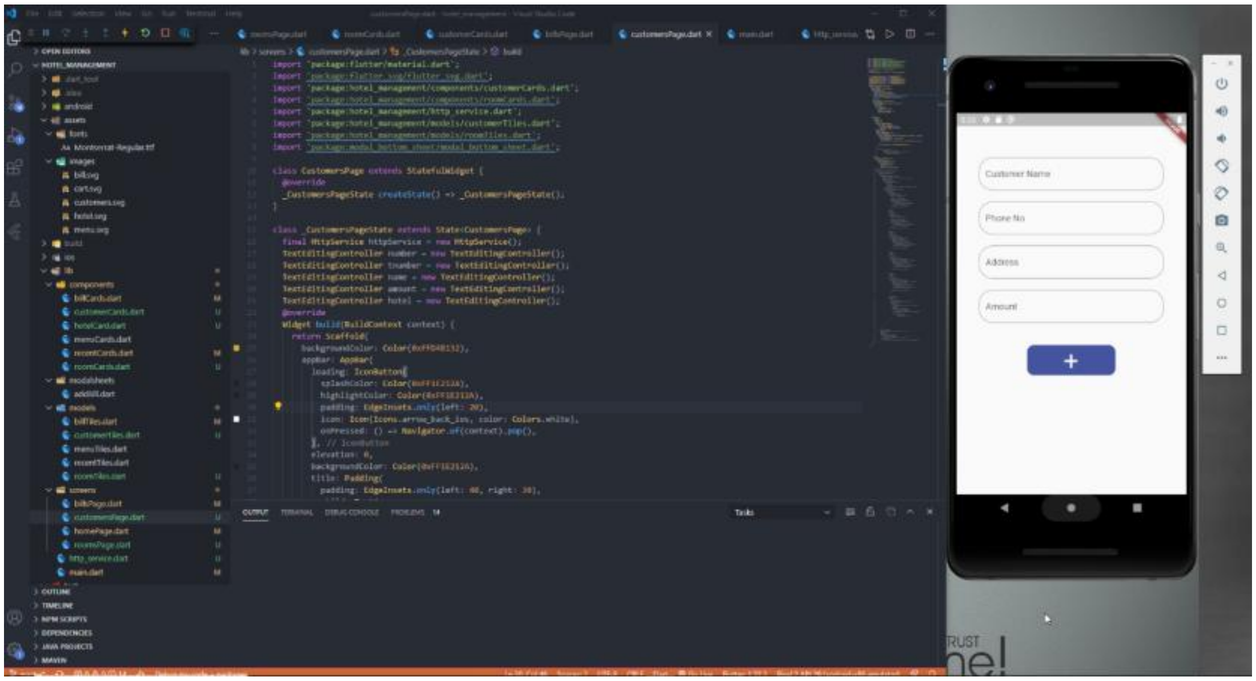


```javascript
const express = require('express');
const bodyParser = require('body-parser');

const HttpError = require('./models/http-error');
const authRoute = require('./routes/authRoute');
const empRoutes = require('./routes/empRoutes');
const adminRoutes = require('./routes/adminRoutes');
const hotelRoutes = require('./routes/hotelRoutes');
const app = express();

app.use(bodyParser.json());

app.use('/api/comp', authRoute);
app.use('/api/emp', empRoutes);
app.use('/api/admin', adminRoutes);
app.use('/api/hotel', hotelRoutes);

app.use((req, res, next) => {
    const error = new HttpError("Could not find this route.", 404);
    throw error;
});

app.use((error, req, res, next) => {
    if (res.headerSent) {
        return next(error);
    }
    res.status(error.code || 500);
    res.json({ message: error.message || "An unknown error occurred!" });
});

app.listen(5000);
```

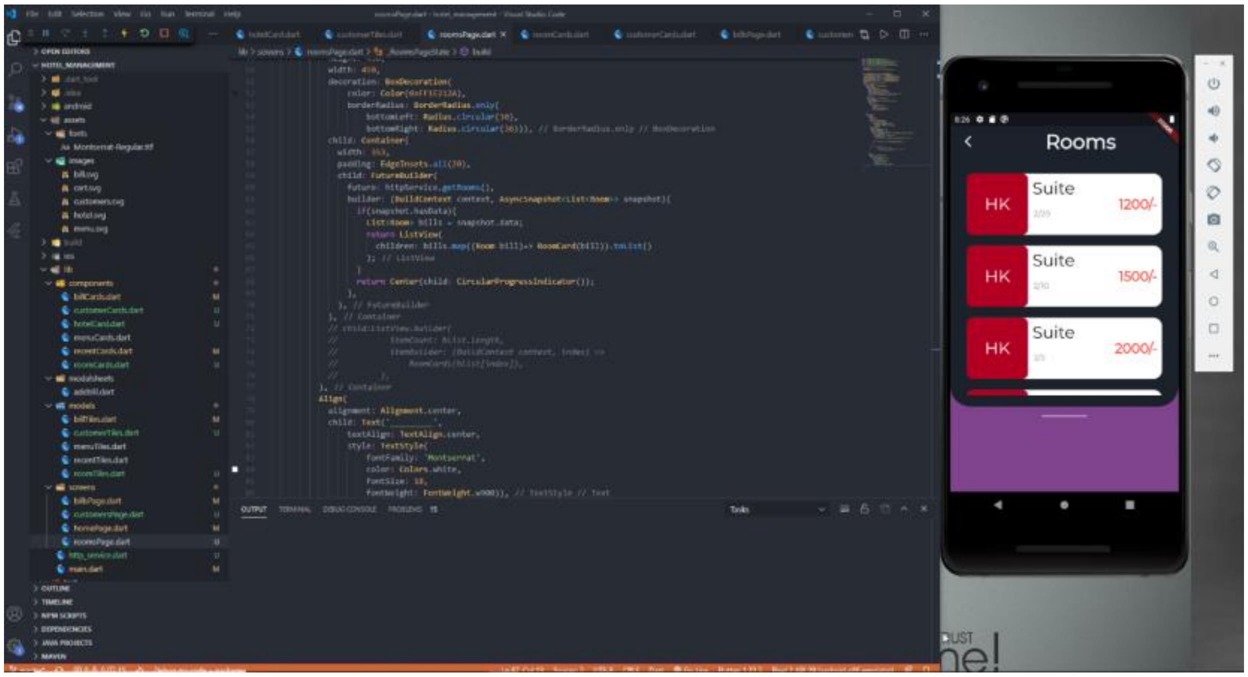
```
{
  _id: 5fa55506811d4883c0c5a820,
  name: 'Harsh',
  phn: 9032185045,
  amount: 20000
}
{
  _id: 5fa5555e811d4883c0c5a822,
  name: 'Bhagi',
  phn: 9784561230,
  amount: 8500
}
```



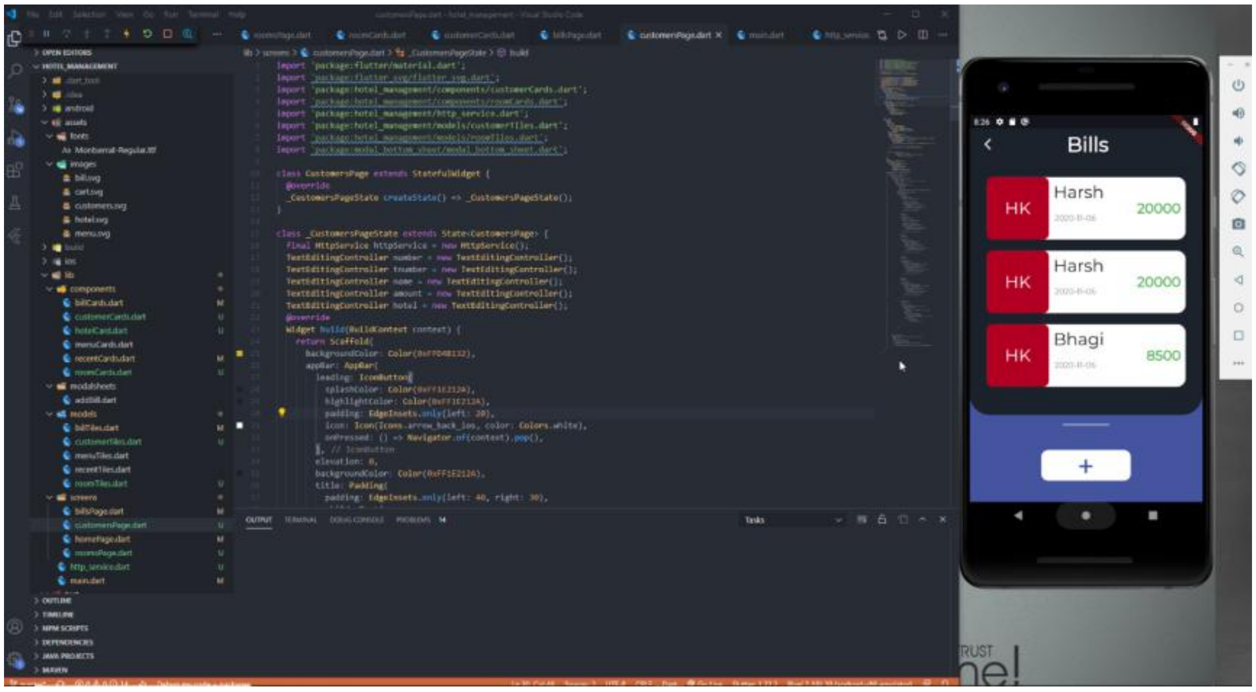
```dart
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:hotel_management/components/customerCards.dart';
import 'package:hotel_management/components/roomCards.dart';
import 'package:hotel_management/http_service.dart';
import 'package:hotel_management/models/customerTiles.dart';
import 'package:hotel_management/models/roomTiles.dart';
import 'package:modal_bottom_sheet/modal_bottom_sheet.dart';

class CustomersPage extends StatefulWidget {
  @override
  _CustomersPageState createState() => _CustomersPageState();
}

class _CustomersPageState extends State<CustomersPage> {
  final HttpService httpService = new HttpService();
  TextEditingController number = new TextEditingController();
  TextEditingController tnumber = new TextEditingController();
  TextEditingController name = new TextEditingController();
  TextEditingController amount = new TextEditingController();
  TextEditingController hotel = new TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFD4B132),
      appBar: AppBar(
        leading: IconButton(
          splashColor: Color(0xFF1E212A),
          highlightColor: Color(0xFF1E212A),
          padding: EdgeInsets.only(left: 20),
          icon: Icon(Icons.arrow_back_ios, color: Colors.white),
          onPressed: () => Navigator.of(context).pop(),
        ), // IconButton
        elevation: 0,
        backgroundColor: Color(0xFF1E212A),
        title: Padding(
          padding: EdgeInsets.only(left: 40, right: 30),
```



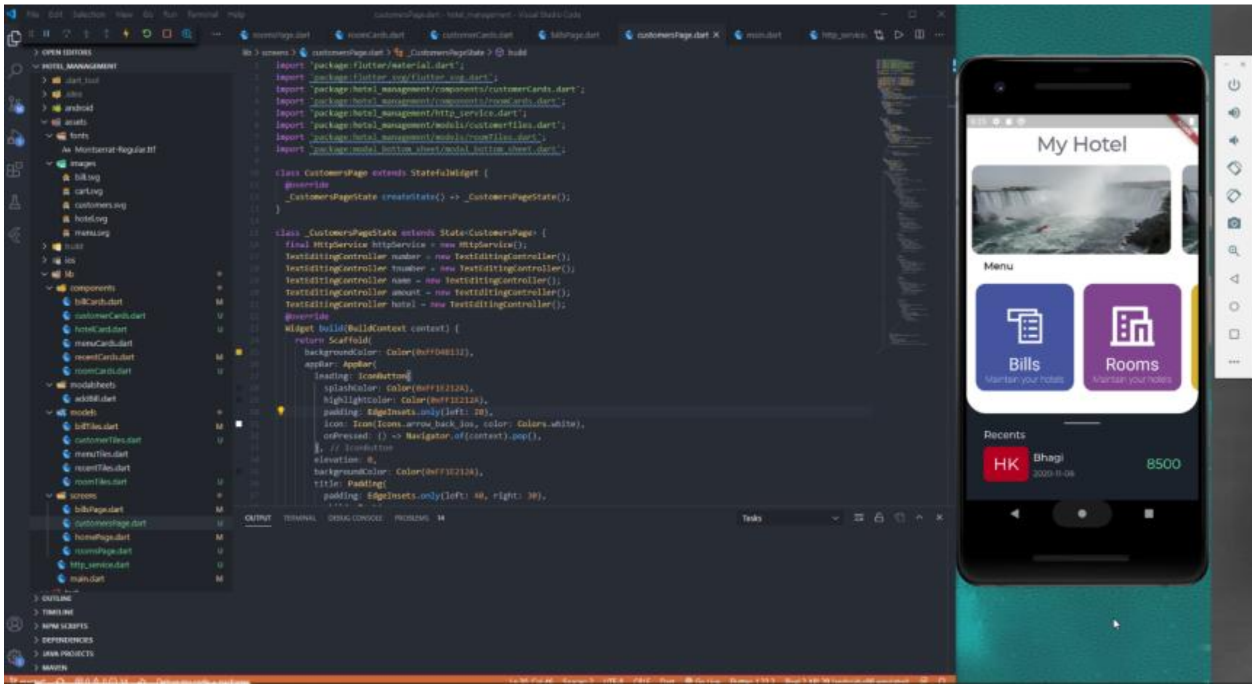
```dart
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:hotel_management/components/customerCards.dart';
import 'package:hotel_management/components/roomCards.dart';
import 'package:hotel_management/http_service.dart';
import 'package:hotel_management/models/customerTiles.dart';
import 'package:hotel_management/models/roomTiles.dart';
import 'package:modal_bottom_sheet/modal_bottom_sheet.dart';

class CustomersPage extends StatefulWidget {
  @override
  _CustomersPageState createState() => _CustomersPageState();
}

class _CustomersPageState extends State<CustomersPage> {
  final HttpService httpService = new HttpService();
  TextEditingController number = new TextEditingController();
  TextEditingController tnumber = new TextEditingController();
  TextEditingController name = new TextEditingController();
  TextEditingController amount = new TextEditingController();
  TextEditingController hotel = new TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFF040132),
      appBar: AppBar(
        leading: IconButton(
          splashColor: Color(0xFF1E212A),
          highlightColor: Color(0xFF1E212A),
          padding: EdgeInsets.only(left: 20),
          icon: Icon(Icons.arrow_back_ios, color: Colors.white),
          onPressed: () => Navigator.of(context).pop(),
        ), // IconButton
        elevation: 0,
        backgroundColor: Color(0xFF1E2126),
        title: Padding(
          padding: EdgeInsets.only(left: 60, right: 30),
```



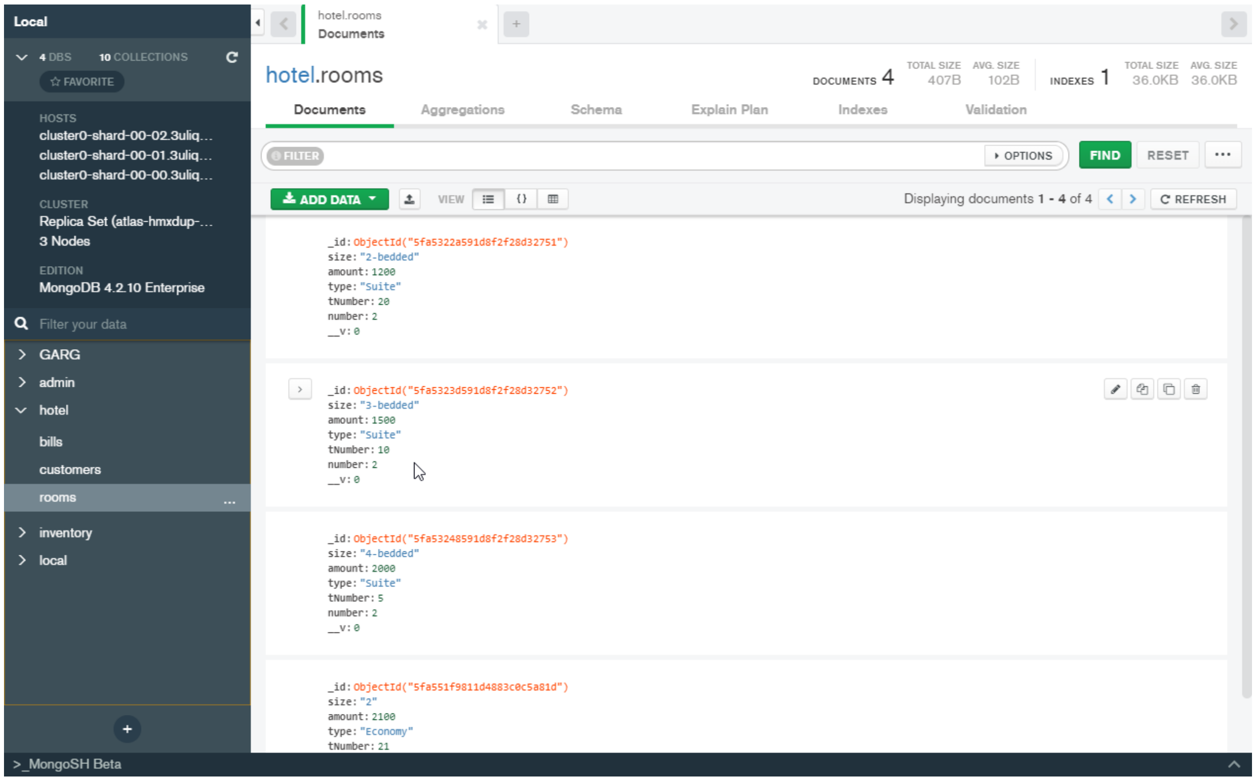
```dart
          width: 410,
          decoration: BoxDecoration(
              color: Color(0xFF1E212A),
              borderRadius: BorderRadius.only(
                  bottomLeft: Radius.circular(30),
                  bottomRight: Radius.circular(30))), // BorderRadius.only // BoxDecoration
          child: Container(
            width: 353,
            padding: EdgeInsets.all(20),
            child: FutureBuilder(
              future: httpService.getRooms(),
              builder: (BuildContext context, AsyncSnapshot<List<Room>> snapshot){
                if(snapshot.hasData){
                  List<Room> bills = snapshot.data;
                  return ListView(
                    children: bills.map((Room bill)=> RoomCard(bill)).toList()
                  ); // ListView
                }
                return Center(child: CircularProgressIndicator());
              },
            ), // FutureBuilder
          ), // Container
          // child:ListView.builder(
          //           itemCount: blist.length,
          //           itemBuilder: (BuildContext context, index) =>
          //               RoomCards(blist[index]),
          //         ),
        ), // Container
        Align(
          alignment: Alignment.center,
          child: Text('__________',
              textAlign: TextAlign.center,
              style: TextStyle(
                  fontFamily: 'Montserrat',
                  color: Colors.white,
                  fontSize: 18,
                  fontWeight: FontWeight.w900)), // TextStyle // Text
```



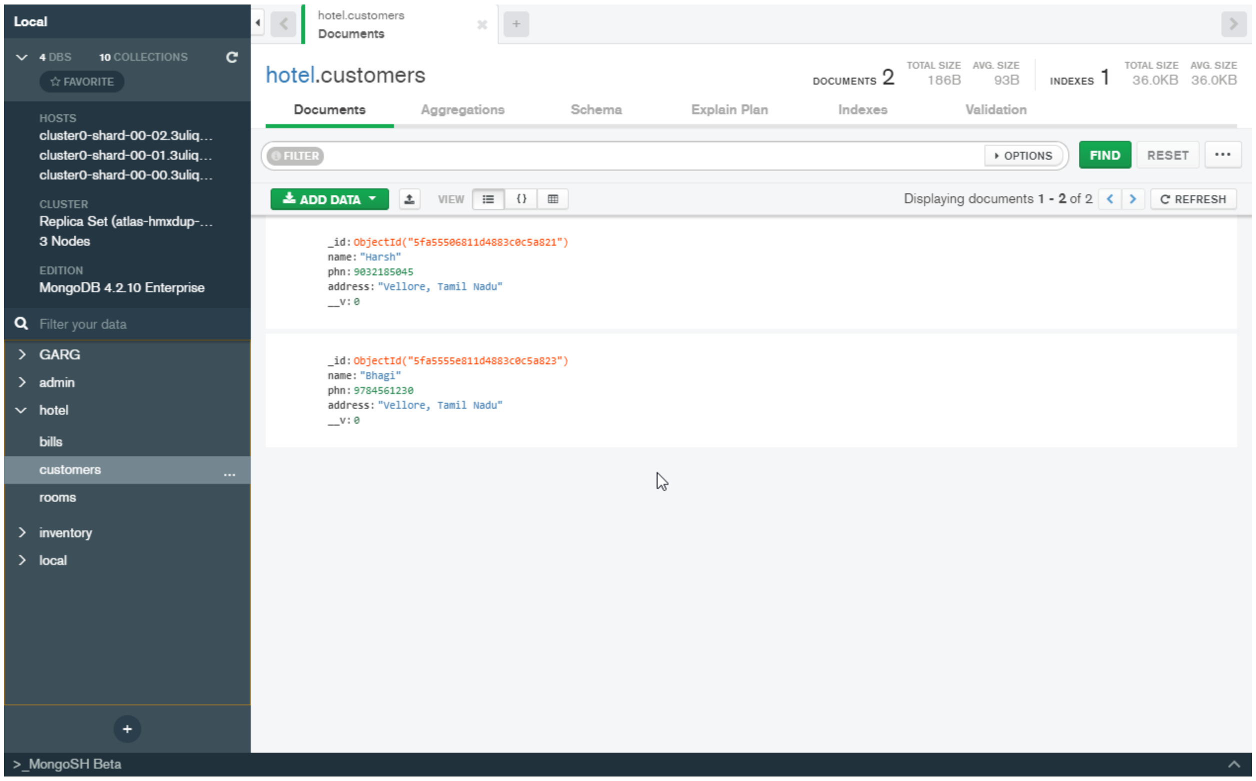
```dart
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:hotel_management/components/customerCards.dart';
import 'package:hotel_management/components/roomCards.dart';
import 'package:hotel_management/http_service.dart';
import 'package:hotel_management/models/customerTiles.dart';
import 'package:hotel_management/models/roomTiles.dart';
import 'package:modal_bottom_sheet/modal_bottom_sheet.dart';

class CustomersPage extends StatefulWidget {
  @override
  _CustomersPageState createState() => _CustomersPageState();
}

class _CustomersPageState extends State<CustomersPage> {
  final HttpService httpService = new HttpService();
  TextEditingController number = new TextEditingController();
  TextEditingController tnumber = new TextEditingController();
  TextEditingController name = new TextEditingController();
  TextEditingController amount = new TextEditingController();
  TextEditingController hotel = new TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFD4B132),
      appBar: AppBar(
        leading: IconButton(
          splashColor: Color(0xFF1E212A),
          highlightColor: Color(0xFF1E212A),
          padding: EdgeInsets.only(left: 20),
          icon: Icon(Icons.arrow_back_ios, color: Colors.white),
          onPressed: () => Navigator.of(context).pop(),
        ), // IconButton
        elevation: 0,
        backgroundColor: Color(0xFF1E212A),
        title: Padding(
          padding: EdgeInsets.only(left: 40, right: 30),
```



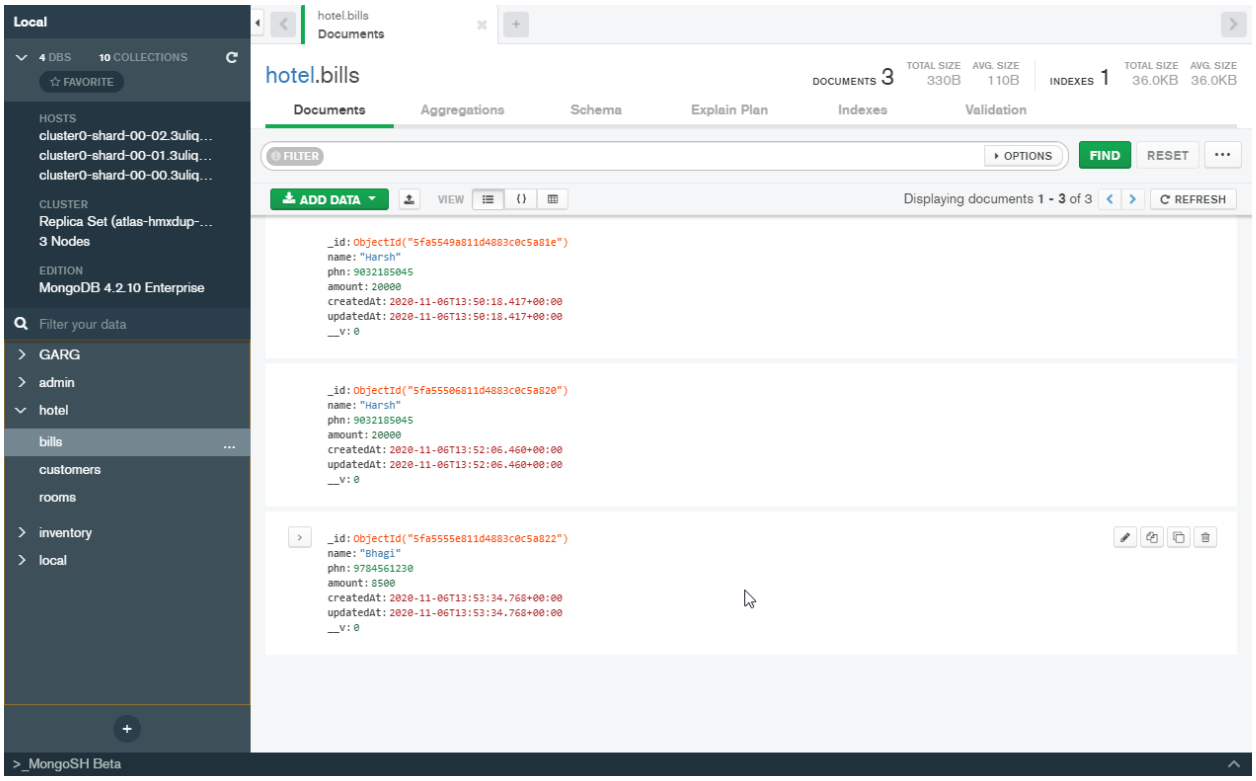
```dart
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:hotel_management/components/customerCards.dart';
import 'package:hotel_management/components/roomCards.dart';
import 'package:hotel_management/http_service.dart';
import 'package:hotel_management/models/customerTiles.dart';
import 'package:hotel_management/models/roomTiles.dart';
import 'package:modal_bottom_sheet/modal_bottom_sheet.dart';

class CustomersPage extends StatefulWidget {
  @override
  _CustomersPageState createState() => _CustomersPageState();
}

class _CustomersPageState extends State<CustomersPage> {
  final HttpService httpService = new HttpService();
  TextEditingController number = new TextEditingController();
  TextEditingController tnumber = new TextEditingController();
  TextEditingController name = new TextEditingController();
  TextEditingController amount = new TextEditingController();
  TextEditingController hotel = new TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFD4B132),
      appBar: AppBar(
        leading: IconButton(
          splashColor: Color(0xFF1E212A),
          highlightColor: Color(0xFF1E212A),
          padding: EdgeInsets.only(left: 20),
          icon: Icon(Icons.arrow_back_ios, color: Colors.white),
          onPressed: () => Navigator.of(context).pop(),
        ), // IconButton
        elevation: 0,
        backgroundColor: Color(0xFF1E212A),
        title: Padding(
          padding: EdgeInsets.only(left: 40, right: 30),
```



```
_id: ObjectId("5fa5322a591d8f2f28d32751")
size: "2-bedded"
amount: 1200
type: "Suite"
tNumber: 20
number: 2
__v: 0

_id: ObjectId("5fa5323d591d8f2f28d32752")
size: "3-bedded"
amount: 1500
type: "Suite"
tNumber: 10
number: 2
__v: 0

_id: ObjectId("5fa53248591d8f2f28d32753")
size: "4-bedded"
amount: 2000
type: "Suite"
tNumber: 5
number: 2
__v: 0

_id: ObjectId("5fa551f9811d4883c0c5a81d")
size: "2"
amount: 2100
type: "Economy"
tNumber: 21
```



```
_id: ObjectId("5fa55506811d4883c0c5a821")
name: "Harsh"
phn: 9032185045
address: "Vellore, Tamil Nadu"
__v: 0

_id: ObjectId("5fa5555e811d4883c0c5a823")
name: "Bhagi"
phn: 9784561230
address: "Vellore, Tamil Nadu"
__v: 0
```



```
_id: ObjectId("5fa5549a811d4883c0c5a81e")
name: "Harsh"
phn: 9032185045
amount: 20000
createdAt: 2020-11-06T13:50:18.417+00:00
updatedAt: 2020-11-06T13:50:18.417+00:00
__v: 0
```

```
_id: ObjectId("5fa55506811d4883c0c5a820")
name: "Harsh"
phn: 9032185045
amount: 20000
createdAt: 2020-11-06T13:52:06.460+00:00
updatedAt: 2020-11-06T13:52:06.460+00:00
__v: 0
```

```
_id: ObjectId("5fa5555e811d4883c0c5a822")
name: "Bhagi"
phn: 9784561230
amount: 8500
createdAt: 2020-11-06T13:53:34.768+00:00
updatedAt: 2020-11-06T13:53:34.768+00:00
__v: 0
```

REFERENCES

During the development of the project, we have used many resources and for that we are grateful to all the people concerned.

Given below are the names of some websites, which we have consulted during the development and documentation of the project.

Websites:

- https://www.google.com/
- https://www.w3schools.com/
- https://stackoverflow.com/
- https://www.geeksforgeeks.org/