Table 1 Contents of the video

Requirement	Start timestamp
1	0s
2	58s
3	2:18s
4	3:45s
5	4:39s
6	5:28s
7	6:11s
8	8:31s
9	10:42s

DBMS Project – Report

- **DBMS used:** PostGreSQL
- **Programming languages used:**
  - Java
  - HTML
  - JavaScript
  - CSS
  - Apache Tomcat
- **API/Libraries used**
  - Google Maps API
    - Integrated maps in our web-app
  - BCrypt
    - Used to securely store customer passwords as hash codes for customer privacy.

Guide to install applications

1. Download IntelliJ
   a. [Download IntelliJ IDEA – The Leading Java and Kotlin IDEJetBrainshttps://www.jetbrains.com › idea › download](#)
2. Download PostGreSQL 16
   a. https://postgresapp.com/downloads.html (MacOS)
   b. https://www.postgresql.org/download/ (General Download link)

3. Download PgAdmin4
   a. https://www.pgadmin.org/download/

List of DDL's

-- ----------------------------

-- Table structure for entities

-- ----------------------------

DROP TABLE IF EXISTS hotelchain;

CREATE TABLE hotelchain (

       HotelChainID serial PRIMARY KEY,

       HotelChainName varchar(45) NOT NULL,

       HotelChainAddress varchar(45) NOT NULL,

       PhoneNumber BIGINT NOT NULL,

       NumberOfHotels INT NOT NULL,

       HotelChainEmail varchar(100) NOT NULL CHECK (HotelChainEmail LIKE

'_%@_%._%')

);

DROP TABLE IF EXISTS hotel;

CREATE TABLE hotel (

       HotelID serial PRIMARY KEY,

       HotelChainID INT REFERENCES hotelchain(HotelChainID),

       HotelPhoneNumber BIGINT NOT NULL,

       Address varchar(45) NOT NULL,

       NumberOfRooms INT NOT NULL,

       HotelCategory varchar(45) NOT NULL,

       HotelArea varchar(45) NOT NULL,

       HotelName varchar(45) NOT NULL,

```sql
        HotelEmail varchar(100) NOT NULL CHECK (HotelEmail LIKE '_%@_%._%'),

        Rating double precision NOT NULL
);


DROP TABLE IF EXISTS room;
CREATE TABLE room (
        RoomID serial PRIMARY KEY,

        HotelID BIGINT REFERENCES hotel(HotelID),

        RoomView varchar(45) NOT NULL,

        Price double precision NOT NULL,

        Capacity INT NOT NULL,

        Amenities TEXT NOT NULL,

        RoomNumber INT NOT NULL,

        Damages TEXT,

        RoomExtension boolean NOT NULL
);
```

```sql
DROP TABLE IF EXISTS customer;
CREATE TABLE customer (
        CustomerID serial PRIMARY KEY,
        CustomerUsername varchar(45) NOT NULL,
        CustomerPassword varchar(255) NOT NULL,
        DateOfRegistration date NOT NULL,
        Address varchar(45) NOT NULL,
        FullName varchar(45) NOT NULL,
        IDPresented varchar(45) NOT NULL,
        CustomerEmail varchar(100) NOT NULL CHECK (CustomerEmail LIKE '_%@_%._%')
);


DROP TABLE IF EXISTS employee;
CREATE TABLE employee (
        EmployeeID serial PRIMARY KEY,
        HotelID serial REFERENCES hotel(HotelID),
        Address varchar(45) NOT NULL,
        FullName varchar(45) NOT NULL,
        EmployeeNumber INT NOT NULL,
        Role varchar(45) NOT NULL
);
DROP TABLE IF EXISTS booking;
CREATE TABLE booking (
        BookingID serial PRIMARY KEY,
        CustomerID BIGINT REFERENCES customer(CustomerID),
        HotelID BIGINT REFERENCES hotel(HotelID),
        RoomID BIGINT REFERENCES room(RoomID),
        Payment boolean DEFAULT false,
        PricePaid double precision NOT NULL,
        CheckIn date NOT NULL,
        CheckOut date NOT NULL);
```

```sql
DROP TABLE IF EXISTS renting;
CREATE TABLE renting (
        RentingID serial PRIMARY KEY,
        CustomerID BIGINT REFERENCES customer(CustomerID),
        RoomID INT REFERENCES room(RoomID),
        CheckIn date NOT NULL,
        CheckOut date NOT NULL
);


DROP TABLE IF EXISTS booking_archive;
CREATE TABLE booking_archive (
    BookingID INT,
    CustomerID INT,
    HotelID INT,
    RoomID INT,
    Payment boolean,
    PricePaid DOUBLE PRECISION NOT NULL,
    CheckIn DATE NOT NULL,
    CheckOut DATE NOT NULL,
    ArchivalTimestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ArchivalReason TEXT,
    PRIMARY KEY (BookingID, ArchivalTimestamp)
);
```

```sql
DROP TABLE IF EXISTS renting_archive;
CREATE TABLE renting_archive (
    RentingID INT,
    CustomerID INT,
    RoomID INT,
    CheckIn DATE NOT NULL,
    CheckOut DATE NOT NULL,
    ArchivalTimestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ArchivalReason TEXT,
    PRIMARY KEY (RentingID, ArchivalTimestamp)
);
```