CSI 2132
Project 2 Report
Student Name: Gisela Rachel Suriawidjaja
Student Number: 300332168

1)

a) PostgreSQL 15
Java: Backend logic using Servlets.
JSP/HTML/CSS/JavaScript: Frontend and User Interface (UI).
Jakarta Servlet (Java EE): To handle HTTP requests and responses.
Apache Tomcat: Server to run and deploy the Java web application.

b)
Download and install PostgreSQL 15.
Download and install Apache Tomcat 10.x+.
Download and install Java JDK 17+.

In PostgreSQL:
CREATE DATABASE hotel_booking;

psql -U postgres -d hotel_booking

Update DatabaseConnection with your PostGreSQL Credentials:

```
public class DatabaseConnection {
    private static final String URL = "jdbc:postgresql://localhost:5432/hotel_booking";
    private static final String USERNAME = "postgres";
    private static final String PASSWORD = "your_password";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USERNAME, PASSWORD);
    }
}
```

Download postgresql-42.x.jar from PostgreSQL JDBC Driver.
Place the .jar file in WEB-INF/lib/.

Place the compiled .war file in Tomcat/webapps/ directory.

Start Apache Tomcat.

Access the application in the browser:

http://localhost:8080/hotel_booking/

Insert all the SQL value from 1c)

c)
-- ==========

```
-- Hotel Chain Table
-- ==========================
CREATE TABLE HotelChain (
chain_id SERIAL PRIMARY KEY,
chain_name VARCHAR(255) UNIQUE NOT NULL,
contact_email VARCHAR(255) NOT NULL CHECK (contact_email LIKE '%@%'),
central_office_address TEXT NOT NULL,
number_of_hotels INT CHECK (number_of_hotels > 0),
phone_number VARCHAR(15) CHECK (phone_number ~ '^[0-9]{10,15}$')
);
-- ==========================
-- Hotel Table
-- ==========================
CREATE TABLE Hotel (
hotel_id SERIAL PRIMARY KEY,
chain_id INT NOT NULL,
category INT CHECK (category BETWEEN 1 AND 5),
phone_number VARCHAR(15) CHECK (phone_number ~ '^[0-9]{10,15}$'),
number_of_rooms INT CHECK (number_of_rooms > 0),
address TEXT NOT NULL,
FOREIGN KEY (chain_id) REFERENCES HotelChain(chain_id) ON DELETE CASCADE
);
-- ==========================
-- Room Table
-- ==========================
CREATE TABLE Room (
room_number INT,
hotel_id INT,
price DECIMAL(10,2) CHECK (price > 0),
capacity INT CHECK (capacity > 0),
view TEXT,
bed_can_be_extended BOOLEAN,
PRIMARY KEY (room_number, hotel_id),
FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id) ON DELETE CASCADE
);
-- ==========================
-- Employee & Role Tables
-- ==========================
CREATE TABLE Employee (
employee_id SERIAL PRIMARY KEY,
full_name VARCHAR(255) NOT NULL,
address TEXT,
SSN_SIN VARCHAR(20) UNIQUE NOT NULL,
email VARCHAR(255) UNIQUE NOT NULL CHECK (email LIKE '%@%.%'),
hotel_id INT NOT NULL,
FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id) ON DELETE CASCADE
);
CREATE TABLE Role (
role_id SERIAL PRIMARY KEY,
```

```
name VARCHAR(255) NOT NULL UNIQUE,
role_description TEXT
);
CREATE TABLE EmployeeRole (
employee_id INT,
role_id INT,
PRIMARY KEY (employee_id, role_id),
FOREIGN KEY (employee_id) REFERENCES Employee(employee_id) ON DELETE CASCADE,
FOREIGN KEY (role_id) REFERENCES Role(role_id) ON DELETE CASCADE
);
-- ==========================
-- Customer Table
-- ==========================
CREATE TABLE Customer (
customer_id SERIAL PRIMARY KEY,
full_name VARCHAR(255) NOT NULL,
type_of_id VARCHAR(50) NOT NULL,
registration_date DATE NOT NULL,
address TEXT
);
-- ==========================
-- Stay, Book, Rent & ArchiveStay Tables
-- ==========================
CREATE TABLE Stay (
stay_id SERIAL PRIMARY KEY,
customer_id INT NULL,
room_number INT NOT NULL,
hotel_id INT NOT NULL,
check_in_date DATE NOT NULL,
check_out_date DATE NOT NULL CHECK (check_out_date > check_in_date),
status VARCHAR(20) CHECK (status IN ('active', 'completed', 'cancelled')) NOT NULL,
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) ON DELETE SET NULL,
FOREIGN KEY (room_number, hotel_id) REFERENCES Room(room_number, hotel_id) ON DELETE
CASCADE
);
CREATE TABLE Book (
booking_id SERIAL PRIMARY KEY,
stay_id INT UNIQUE,
booking_date DATE NOT NULL,
FOREIGN KEY (stay_id) REFERENCES Stay(stay_id) ON DELETE CASCADE
);
CREATE TABLE Rent (
rent_id SERIAL PRIMARY KEY,
stay_id INT UNIQUE,
booking_id INT UNIQUE NULL,
employee_id INT NOT NULL,
payment_status VARCHAR(20) CHECK (payment_status IN ('pending', 'paid')) NOT NULL,
FOREIGN KEY (stay_id) REFERENCES Stay(stay_id) ON DELETE CASCADE,
FOREIGN KEY (booking_id) REFERENCES Book(booking_id) ON DELETE CASCADE,
```

```
FOREIGN KEY (employee_id) REFERENCES Employee(employee_id) ON DELETE SET NULL
);
CREATE TABLE ArchiveStay (
archive_id SERIAL PRIMARY KEY,
stay_id INT UNIQUE,
customer_name VARCHAR(255),
room_number INT,
hotel_id INT,
check_in_date DATE NOT NULL,
check_out_date DATE NOT NULL CHECK (check_out_date > check_in_date),
status VARCHAR(20) CHECK (status IN ('completed', 'cancelled')) NOT NULL,
FOREIGN KEY (stay_id) REFERENCES Stay(stay_id) ON DELETE CASCADE
);
-- ==========================
-- Problem Reports Table
-- ==========================
CREATE TABLE Problems (
problem_id SERIAL PRIMARY KEY,
room_number INT NOT NULL,
hotel_id INT NOT NULL,
description TEXT NOT NULL,
reported_date DATE NOT NULL,
fixed_date DATE,
status VARCHAR(20) CHECK (status IN ('pending', 'in progress', 'resolved')) NOT NULL,
FOREIGN KEY (room_number, hotel_id) REFERENCES Room(room_number, hotel_id) ON DELETE
CASCADE
);
-- ==========================
-- Amenities & Features Tables
-- ==========================
CREATE TABLE Amenities (
amenity_id SERIAL PRIMARY KEY,
name VARCHAR(255) NOT NULL UNIQUE,
description TEXT
);
CREATE TABLE Features (
room_number INT,
hotel_id INT,
amenity_id INT,
PRIMARY KEY (room_number, hotel_id, amenity_id),
FOREIGN KEY (room_number, hotel_id) REFERENCES Room(room_number, hotel_id) ON DELETE
CASCADE,
FOREIGN KEY (amenity_id) REFERENCES Amenities(amenity_id) ON DELETE CASCADE
);
INSERT INTO HotelChain (chain_id, chain_name, contact_email, central_office_address,
number_of_hotels, phone_number) VALUES
(1, 'Hilton Hotels', 'contact@hilton.com', '7930 Jones Branch Dr, McLean, VA, USA', 8, '1234567890'),
(2, 'Marriott International', 'contact@marriott.com', '10400 Fernwood Rd, Bethesda, MD, USA', 8,
'2345678901'),
```

```
(3, 'Hyatt Hotels', 'contact@hyatt.com', '150 N Riverside Plaza, Chicago, IL, USA', 8, '3456789012'),
(4, 'Wyndham Hotels', 'contact@wyndham.com', '22 Sylvan Way, Parsippany, NJ, USA', 8, '4567890123'),
(5, 'Four Seasons', 'contact@fourseasons.com', '1165 Leslie Street, Toronto, ON, Canada', 8,
'5678901234');
-- Insert Hotels AFTER Hotel Chains
INSERT INTO Hotel (hotel_id, chain_id, category, phone_number, number_of_rooms, address) VALUES
(1, 1, 5, '1112223333', 120, 'Hilton Toronto, Toronto, ON, Canada'),
(2, 1, 4, '1112223334', 150, 'Hilton Vancouver, Vancouver, BC, Canada'),
(3, 1, 3, '1112223335', 110, 'Hilton Los Angeles, Los Angeles, CA, USA'),
(4, 1, 5, '1112223336', 200, 'Hilton New York, NY, USA'),
(5, 2, 5, '2112223333', 160, 'Marriott Toronto, Toronto, ON, Canada'),
(6, 2, 3, '2112223334', 140, 'Marriott Vancouver, Vancouver, BC, Canada'),
(7, 2, 2, '2112223335', 100, 'Marriott Montreal, QC, Canada'),
(8, 3, 4, '3112223336', 180, 'Hyatt Boston, MA, USA');
-- Insert Rooms AFTER Hotels
INSERT INTO Room (room_number, hotel_id, price, capacity, view, bed_can_be_extended) VALUES
(101, 1, 200, 1, 'City View', FALSE),
(102, 1, 250, 2, 'Sea View', TRUE),
(103, 1, 300, 3, 'Mountain View', FALSE),
(104, 1, 350, 4, 'Sea View', TRUE),
(105, 1, 400, 5, 'City View', FALSE),
(101, 2, 180, 1, 'City View', FALSE),
(102, 2, 220, 2, 'Sea View', TRUE),
(103, 2, 270, 3, 'Mountain View', FALSE),
(104, 2, 320, 4, 'Sea View', TRUE),
(105, 2, 370, 5, 'City View', FALSE);
-- Insert Amenities BEFORE Features
INSERT INTO Amenities (amenity_id, name, description) VALUES
(1, 'Wi-Fi', 'Free high-speed internet'),
(2, 'TV', 'Flat-screen television with cable'),
(3, 'Mini Fridge', 'Mini fridge for guest use'),
(4, 'Air Conditioner', 'Temperature control unit'),
(5, 'Safe', 'Secure storage for valuables');
-- Insert Features AFTER Amenities and Rooms
INSERT INTO Features (room_number, hotel_id, amenity_id) VALUES
(101, 1, 1), (102, 1, 2), (103, 1, 3), (104, 1, 4), (105, 1, 5),
(101, 2, 1), (102, 2, 2), (103, 2, 3), (104, 2, 4), (105, 2, 5);
-- Insert Employees AFTER Hotels
INSERT INTO Employee (employee_id, full_name, address, SSN_SIN, email, hotel_id) VALUES
(1, 'John Smith', 'Toronto, ON, Canada', '111-22-3333', 'john.smith@hilton.com', 1),
(2, 'Sarah Johnson', 'Vancouver, BC, Canada', '222-33-4444', 'sarah.johnson@hilton.com', 2);
-- Insert Roles BEFORE Assigning Employee Roles
INSERT INTO Role (role_id, name, role_description) VALUES
(1, 'Manager', 'Manages hotel operations');
-- Assign Employee Roles AFTER Roles Exist
INSERT INTO EmployeeRole (employee_id, role_id) VALUES
(1, 1), (2, 1);
-- Insert Customers BEFORE Stays
INSERT INTO Customer (customer_id, full_name, type_of_id, registration_date, address) VALUES
```

```
(1, 'Alice Carter', 'Passport', '2024-01-01', 'New York, NY, USA'),
(2, 'Bob Anderson', 'Driver License', '2024-02-15', 'Chicago, IL, USA');
-- Insert Stays AFTER Customers and Rooms Exist
INSERT INTO Stay (stay_id, customer_id, room_number, hotel_id, check_in_date, check_out_date,
status) VALUES
(1, 1, 101, 1, '2025-03-10', '2025-03-15', 'active'),
(2, 2, 102, 2, '2025-02-05', '2025-02-10', 'completed');
-- Insert Bookings AFTER Stays Exist
INSERT INTO Book (booking_id, stay_id, booking_date) VALUES
(1, 1, '2025-02-15'),
(2, 2, '2025-01-01');
-- Insert Rents AFTER Bookings Exist
INSERT INTO Rent (rent_id, stay_id, booking_id, employee_id, payment_status) VALUES
(1, 1, 1, 1, 'paid'),
(2, 2, 2, 2, 'paid');
-- Insert ArchiveStay AFTER Completed Stay Exists
INSERT INTO ArchiveStay (archive_id, stay_id, customer_name, room_number, hotel_id, check_in_date,
check_out_date, status) VALUES
(1, 2, 'Bob Anderson', 102, 2, '2025-02-05', '2025-02-10', 'completed');
-- Insert Problems AFTER Rooms Exist
INSERT INTO Problems (problem_id, room_number, hotel_id, description, reported_date, fixed_date,
status) VALUES
(1, 101, 1, 'Air conditioner not working', '2025-02-10', NULL, 'pending'),
(2, 102, 2, 'TV not functioning', '2025-01-05', '2025-01-10', 'resolved');
-- Insert Additional Hotels to Ensure Each Chain Has at Least 8 Hotels
INSERT INTO Hotel (hotel_id, chain_id, category, phone_number, number_of_rooms, address) VALUES
-- Additional Hilton Hotels
(9, 1, 3, '1112223341', 130, 'Hilton Chicago, Chicago, IL, USA'),
(10, 1, 2, '1112223342', 120, 'Hilton Miami, FL, USA'),
(11, 1, 4, '1112223343', 140, 'Hilton Dallas, TX, USA'),
(12, 1, 5, '1112223344', 160, 'Hilton San Francisco, CA, USA'),
-- Additional Marriott Hotels
(13, 2, 4, '2112223341', 170, 'Marriott Los Angeles, CA, USA'),
(14, 2, 3, '2112223342', 150, 'Marriott New York, NY, USA'),
(15, 2, 5, '2112223343', 190, 'Marriott Chicago, IL, USA'),
(16, 2, 4, '2112223344', 180, 'Marriott Miami, FL, USA'),
(17, 2, 3, '2112223345', 160, 'Marriott San Francisco, CA, USA'),
-- Additional Hyatt Hotels
(18, 3, 5, '3112223341', 200, 'Hyatt Toronto, Toronto, ON, Canada'),
(19, 3, 3, '3112223342', 140, 'Hyatt Vancouver, BC, Canada'),
(20, 3, 2, '3112223343', 110, 'Hyatt Montreal, QC, Canada'),
(21, 3, 4, '3112223344', 150, 'Hyatt Dallas, TX, USA'),
(22, 3, 5, '3112223345', 170, 'Hyatt Washington, DC, USA'),
(23, 3, 3, '3112223346', 130, 'Hyatt Seattle, WA, USA'),
(24, 3, 2, '3112223347', 120, 'Hyatt Denver, CO, USA'),
-- Additional Four Seasons Hotels
(25, 5, 5, '5112223333', 210, 'Four Seasons Toronto, Toronto, ON, Canada'),
(26, 5, 4, '5112223334', 170, 'Four Seasons Vancouver, BC, Canada'),
(27, 5, 3, '5112223335', 140, 'Four Seasons Montreal, QC, Canada'),
```

```
(28, 5, 2, '5112223336', 120, 'Four Seasons Los Angeles, CA, USA'),
(29, 5, 5, '5112223337', 230, 'Four Seasons New York, NY, USA'),
(30, 5, 3, '5112223338', 150, 'Four Seasons Miami, FL, USA'),
(31, 5, 4, '5112223339', 190, 'Four Seasons San Francisco, CA, USA'),
(32, 5, 5, '5112223340', 220, 'Four Seasons Chicago, IL, USA'),
-- Additional Wyndham Hotels
(33, 4, 3, '4112223333', 150, 'Wyndham Toronto, Toronto, ON, Canada'),
(34, 4, 2, '4112223334', 130, 'Wyndham Vancouver, BC, Canada'),
(35, 4, 4, '4112223335', 170, 'Wyndham Montreal, QC, Canada'),
(36, 4, 5, '4112223336', 200, 'Wyndham Los Angeles, CA, USA'),
(37, 4, 3, '4112223337', 140, 'Wyndham New York, NY, USA'),
(38, 4, 2, '4112223338', 120, 'Wyndham Miami, FL, USA'),
(39, 4, 4, '4112223339', 180, 'Wyndham San Francisco, CA, USA'),
(40, 4, 5, '4112223340', 210, 'Wyndham Chicago, IL, USA');
-- Insert Additional Rooms for Marriott
INSERT INTO Room (room_number, hotel_id, price, capacity, view, bed_can_be_extended) VALUES
(106, 5, 380, 1, 'City View', FALSE),
(107, 5, 420, 2, 'Sea View', TRUE);
-- Insert Additional Rooms for Hyatt (5 more rooms)
INSERT INTO Room (room_number, hotel_id, price, capacity, view, bed_can_be_extended) VALUES
(101, 18, 210, 1, 'Garden View', FALSE),
(102, 18, 260, 2, 'City View', TRUE),
(103, 18, 310, 3, 'Mountain View', FALSE),
(104, 18, 360, 4, 'Sea View', TRUE),
(105, 18, 410, 5, 'City View', FALSE);
-- Insert Additional Rooms for Four Seasons (5 more rooms)
INSERT INTO Room (room_number, hotel_id, price, capacity, view, bed_can_be_extended) VALUES
(101, 25, 250, 1, 'City View', FALSE),
(102, 25, 300, 2, 'Sea View', TRUE),
(103, 25, 350, 3, 'Mountain View', FALSE),
(104, 25, 400, 4, 'Sea View', TRUE),
(105, 25, 450, 5, 'City View', FALSE);
-- Insert Additional Rooms for Wyndham (5 more rooms)
INSERT INTO Room (room_number, hotel_id, price, capacity, view, bed_can_be_extended) VALUES
(101, 33, 190, 1, 'Garden View', FALSE),
(102, 33, 240, 2, 'City View', TRUE),
(103, 33, 290, 3, 'Mountain View', FALSE),
(104, 33, 340, 4, 'Sea View', TRUE),
(105, 33, 390, 5, 'City View', FALSE);
```

2c

```
SELECT r.hotel_id, h.address, MAX(r.price) AS max_room_price
FROM Room r
JOIN Hotel h ON r.hotel_id = h.hotel_id
GROUP BY r.hotel_id, h.address
ORDER BY max_room_price DESC;
SELECT hc.chain_name, COUNT(h.hotel_id) AS total_hotels
FROM HotelChain hc
LEFT JOIN Hotel h ON hc.chain_id = h.chain_id
GROUP BY hc.chain_name
```

```
ORDER BY total_hotels DESC;
SELECT h.hotel_id, h.address, h.number_of_rooms
FROM Hotel h
WHERE h.number_of_rooms > (
   SELECT AVG(number_of_rooms) FROM Hotel
)
ORDER BY h.number_of_rooms DESC;
SELECT hotel_id, address, phone_number, category, number_of_rooms
FROM Hotel
WHERE address LIKE '%Toronto%'
ORDER BY category DESC;
```

2d

```
CREATE OR REPLACE FUNCTION check_room_availability()
RETURNS TRIGGER AS $$
BEGIN
   IF EXISTS (
      SELECT 1 FROM Stay
      WHERE room_number = NEW.room_number
      AND hotel_id = NEW.hotel_id
      AND status = 'active'
      AND (
         (NEW.check_in_date BETWEEN check_in_date AND check_out_date) OR
         (NEW.check_out_date BETWEEN check_in_date AND check_out_date) OR
         (check_in_date BETWEEN NEW.check_in_date AND NEW.check_out_date)
      )
   ) THEN
      RAISE EXCEPTION 'Room % in Hotel % is already booked for the selected dates.',
         NEW.room_number, NEW.hotel_id;
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER trigger_check_room_availability
BEFORE INSERT ON Stay
FOR EACH ROW
EXECUTE FUNCTION check_room_availability();
CREATE OR REPLACE FUNCTION update_hotel_count()
RETURNS TRIGGER AS $$
BEGIN
   IF TG_OP = 'INSERT' THEN
      UPDATE HotelChain
      SET number_of_hotels = number_of_hotels + 1
      WHERE chain_id = NEW.chain_id;
   ELSIF TG_OP = 'DELETE' THEN
      UPDATE HotelChain
      SET number_of_hotels = number_of_hotels - 1
      WHERE chain_id = OLD.chain_id;
   END IF;
   RETURN NULL;
```

```
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER trigger_update_hotel_count
AFTER INSERT OR DELETE ON Hotel
FOR EACH ROW
EXECUTE FUNCTION update_hotel_count();
```

2e

```
CREATE INDEX idx_stay_room_dates
ON Stay (room_number, hotel_id, check_in_date, check_out_date);
CREATE INDEX idx_hotel_chain_category
ON Hotel (chain_id, category);
CREATE INDEX idx_customer_name_id
ON Customer (full_name, type_of_id);
```

2f

```
CREATE VIEW AvailableRoomsPerArea AS
SELECT
    h.address AS area,
    COUNT(r.room_number) AS available_rooms
FROM Room r
JOIN Hotel h ON r.hotel_id = h.hotel_id
LEFT JOIN Stay s ON r.room_number = s.room_number AND r.hotel_id = s.hotel_id
    AND s.status = 'active'
WHERE s.stay_id IS NULL -- Rooms not currently booked or rented
GROUP BY h.address;
CREATE VIEW HotelTotalRoomCapacity AS
SELECT
    h.hotel_id,
    h.address AS hotel_name,
    SUM(r.capacity) AS total_capacity
FROM Room r
JOIN Hotel h ON r.hotel_id = h.hotel_id
GROUP BY h.hotel_id, h.address;
```