

AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

Faculty of Science And Technology

Finalterm Report Cover Sheet



Students must complete all details except the faculty use part.

Title: Paying Guest Management System

Due Date: 06-01-2024 Semester: Fall 23-24 Degree Program: CSE

Subject Name: ADVANCE DATABASE MANAGEMENT SYSTEM Section: B

Course Instructor: Rezwan Ahmed

Declaration and Statement of Authorship:

1. I/we hold a copy of this report, which can be produced if the original is lost/ damaged.
2. This report is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this report has been written for me/us by any other person except where such collaboration has been authorized by the lecturer/teacher concerned and is clearly acknowledged in the report.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the School for review and comparison, including review by external examiners.

Group Number (if applicable): ☐ Individual Submission ☒ Group Submission

No.	Student Name	Student Number	Student Signature	Date
Group Members:				
1	Md. Shariar Hosain Sanny	21-44677-1		06-01-24
2	Shariar Rahman Apurbo	21-44738-1		06-01-24
3	Sanjida Jahan	21-44684-1		06-01-24

1) VIEW

1.1

```
CREATE OR REPLACE FORCE VIEW "PAYING-GUEST-PROJECT"."BOOKING_VIEW" ("BOOKING_ID", "POST_ID", "USERNAME", "BOOKING_APPROVAL", "DATE_OF_BOOKING", "NAME", "EMAIL", "PHONE", "HOUSE_NUMBER", "NUMBER_OF_ROOMS", "PRAFAREBLE_GENDER", "RENT", "ADDRESS", "BOOKING_STATUS", "DATE_OF_POST") AS
SELECT
booking."BOOKING_ID", booking."POST_ID", booking."USERNAME", booking."BOOKING_APPROVAL", booking."DATE_OF_BOOKING", guestregistration.name, guestregistration.email, guestregistration.phone, post.House_Number, post.Number_of_Rooms, post.Prafareble_Gender, post.Rent, post.Address, post.Booking_Status, post.Date_of_Post
FROM booking
JOIN guestregistration ON booking.username = guestregistration.username
JOIN post ON booking.post_id = post.post_id
WHERE booking.booking_approval = 'requested';
```

1.2

```
CREATE OR REPLACE VIEW all post AS
SELECT * FROM post;
```

2) PROCEDURES AND FUNCTIONS

2.1

```
CREATE OR REPLACE PACKAGE booking_package AS
    PROCEDURE set_booking_approval_admin(p_post_id booking.post_id%TYPE);
    FUNCTION is_update_successful RETURN BOOLEAN;
END booking_package;
CREATE OR REPLACE PACKAGE BODY booking_package AS
    g_rows_updated NUMBER := 0;

    PROCEDURE set_booking_approval_admin(p_post_id booking.post_id%TYPE) IS
    BEGIN
        UPDATE booking SET booking_approval = 'approved' WHERE post_id = p_post_id;
        g_rows_updated := SQL%ROWCOUNT;
    END set_booking_approval_admin;

    FUNCTION is_update_successful RETURN BOOLEAN IS
    BEGIN
        IF g_rows_updated > 0 THEN
            RETURN TRUE;
        ELSE
```

```

        RETURN FALSE;
    END IF;
    END is_update_successful;
END booking_package;

```

2.2

```

CREATE OR REPLACE PACKAGE chekin_package AS
    TYPE refcur IS REF CURSOR;

    PROCEDURE show_info_forcheckin(p_booking_id IN booking.booking_id%TYPE,
    p_refcur OUT refcur);
    PROCEDURE insert_checkin(p_booking_id IN booking.booking_id%TYPE,
    p_checkin_status IN chekin_checkout.checkin_status%TYPE, p_checkout_status
    IN chekin_checkout.checkout_status%TYPE);
    PROCEDURE update_booking_status(p_booking_id IN
    booking.booking_id%TYPE);
END chekin_package ;

CREATE OR REPLACE PACKAGE BODY chekin_package AS
    PROCEDURE show_info_forcheckin(p_booking_id IN booking.booking_id%TYPE,
    p_refcur OUT refcur) IS
    BEGIN
        OPEN p_refcur FOR
        SELECT booking.*,
        guestregistration.name,guestregistration.email,guestregistration.phone,post.Hous
        e_Number,post.Number_of_Rooms,post.Prafareble_Gender,post.Rent,post.Add
        ress,post.Booking_Status,post.Date_of_Post
        FROM booking,guestregistration,post
        WHERE booking.username=guestregistration.username
        AND booking.post_id=post.post_id AND booking.booking_id=p_booking_id;
    END show_info_forcheckin;

    PROCEDURE insert_checkin(p_booking_id IN booking.booking_id%TYPE,
    p_checkin_status IN chekin_checkout.checkin_status%TYPE, p_checkout_status
    IN chekin_checkout.checkout_status%TYPE) IS
        v_id chekin_checkout.id%TYPE;
        v_checkin_date chekin_checkout.checkin_date%TYPE := SYSDATE;
        v_checkout_date chekin_checkout.checkout_date%TYPE := NULL;
    BEGIN
        SELECT chekin_checkout_id_seq.NEXTVAL INTO v_id FROM dual;
        INSERT INTO chekin_checkout (id, booking_id, checkin_status, checkin_date,
        checkout_status, checkout_date) VALUES (v_id, p_booking_id, p_checkin_status,
        v_checkin_date, p_checkout_status, v_checkout_date);
    END insert_checkin;

    PROCEDURE update_booking_status(p_booking_id IN
    booking.booking_id%TYPE) IS

```

```

BEGIN
    UPDATE booking SET booking_approval='checkin' WHERE
    booking_id=p_booking_id;
    END update_booking_status;
END chekin_package ;

```

2.3

```

CREATE OR REPLACE PROCEDURE post_ownerinformation(p_post_id IN
post.post_id%TYPE, p_refcur OUT SYS_REFCURSOR) IS
BEGIN
    OPEN p_refcur FOR
    SELECT post.*, hostregistration.*
    FROM post, hostregistration
    WHERE post.username = hostregistration.username
    AND post.post_id = p_post_id;
END post_ownerinformation;

```

2.4

```

CREATE OR REPLACE PROCEDURE selectgenderby_username(p_username IN
guestregistration.username%TYPE, p_refcur OUT SYS_REFCURSOR) IS
BEGIN
    OPEN p_refcur FOR
    SELECT gender FROM guestregistration WHERE username = p_username;
END selectgenderby_username;

```

2.5(searching)

```

CREATE OR REPLACE PROCEDURE filterByPrice(p_price IN post.Rent%TYPE, p_refcur
OUT SYS_REFCURSOR) IS
BEGIN
    OPEN p_refcur FOR
    SELECT * FROM post WHERE Rent >= p_price ORDER BY CASE WHEN Rent = p_price
    THEN 1 ELSE 0 END DESC, Rent ASC;
END filterByPrice;

```

3) Tigger

3.1

```
CREATE OR REPLACE TRIGGER check_phone_before_insert
BEFORE INSERT ON GUESTREGISTRATION
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM GUESTREGISTRATION WHERE phone =
:NEW.phone;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, ' Phone number already exists');
    END IF;
END check_phone_before_insert;
```

3.2

```
CREATE OR REPLACE TRIGGER check_username_before_insert
BEFORE INSERT ON GUESTREGISTRATION
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM GUESTREGISTRATION WHERE username
= :NEW.username;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Error: Username already exists.');
```

3.3

```
CREATE OR REPLACE TRIGGER check_history_id
BEFORE INSERT ON billing
FOR EACH ROW
DECLARE
    history_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO history_count FROM billing WHERE history_id =
:new.history_id;
    IF history_count > 0 THEN
```

```
        RAISE_APPLICATION_ERROR(-20001, 'History ID already exists in the billing table.');
```

```
    END IF;
```

```
END;
```

3.4(sequence with Tigger)

```
CREATE SEQUENCE post_id_seq
START WITH 1
INCREMENT BY 1;

CREATE OR REPLACE TRIGGER post_id_trigger
BEFORE INSERT ON post
FOR EACH ROW
BEGIN
    SELECT post_id_seq.NEXTVAL
    INTO :new.POST_ID
    FROM dual;
END;
```

3.5(sequence with Tigger)

```
CREATE SEQUENCE billing_id_seq
START WITH 1
INCREMENT BY 1;

CREATE OR REPLACE TRIGGER billing_id_trigger
BEFORE INSERT ON billing
FOR EACH ROW
BEGIN
    SELECT billing_id_seq.NEXTVAL
    INTO :new.id
    FROM dual;
END;
```

4) SEQUENCE

```
CREATE SEQUENCE booking_id_seq
START WITH 1
INCREMENT BY 1;
```

```
CREATE SEQUENCE chekin_checkout_id_seq  
  START WITH 15  
  INCREMENT BY 1;
```

```
CREATE SEQUENCE history_id_seq START WITH 20 INCREMENT BY 1;
```