

# DBMS: Railway Reservation

---

Deeksha Singh Duvesh(2020049),  
Chaitanya jain(2020044),  
Mohit (2020085)



INDRAPRASTHA INSTITUTE *of*  
INFORMATION TECHNOLOGY **DELHI**



# Individual efforts

---

***Chaitanya Jain*** : SQL Queries, Embedded SQL Queries, Query optimization, Triggers

***Mohit*** : SQL Queries, Embedded SQL Queries, Query optimization, Triggers

***Deeksha Singh Duvesh*** : Views and Grants, Project documentation, Project Scope and Relational Schema, Indexing

# Problem Statement

---

We often face issues with getting proper information from the authorities during train travel like we don't know the proper exact train timings, or how much long is the waiting list. The authorities also face issues managing all these many passengers on paper. So, we thought of making a portal/system that would cater to all these needs and make a passengers journey pleasurable. We have added admin side portal too for storing and showing details for their perusal too.

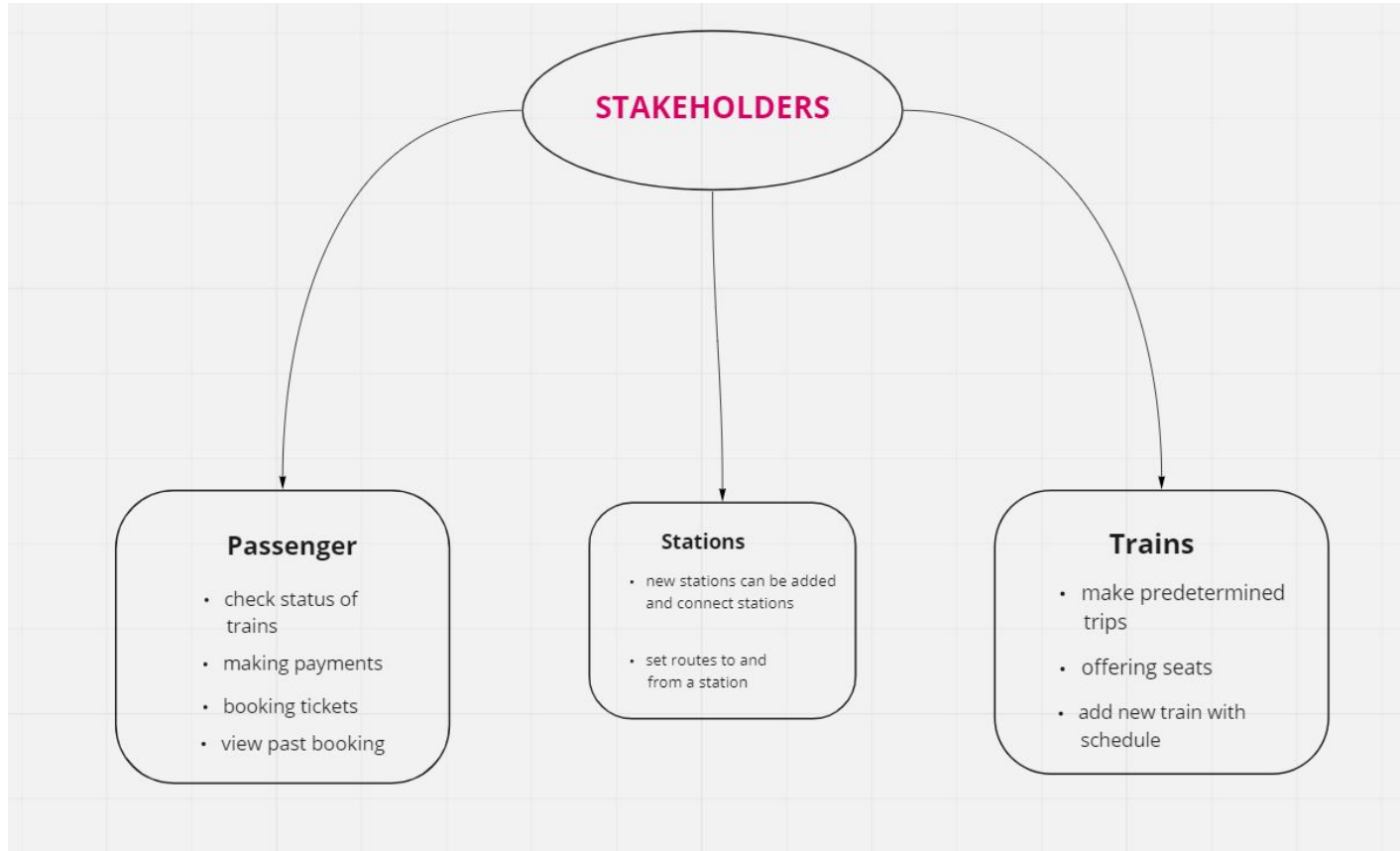
# Scope:

---

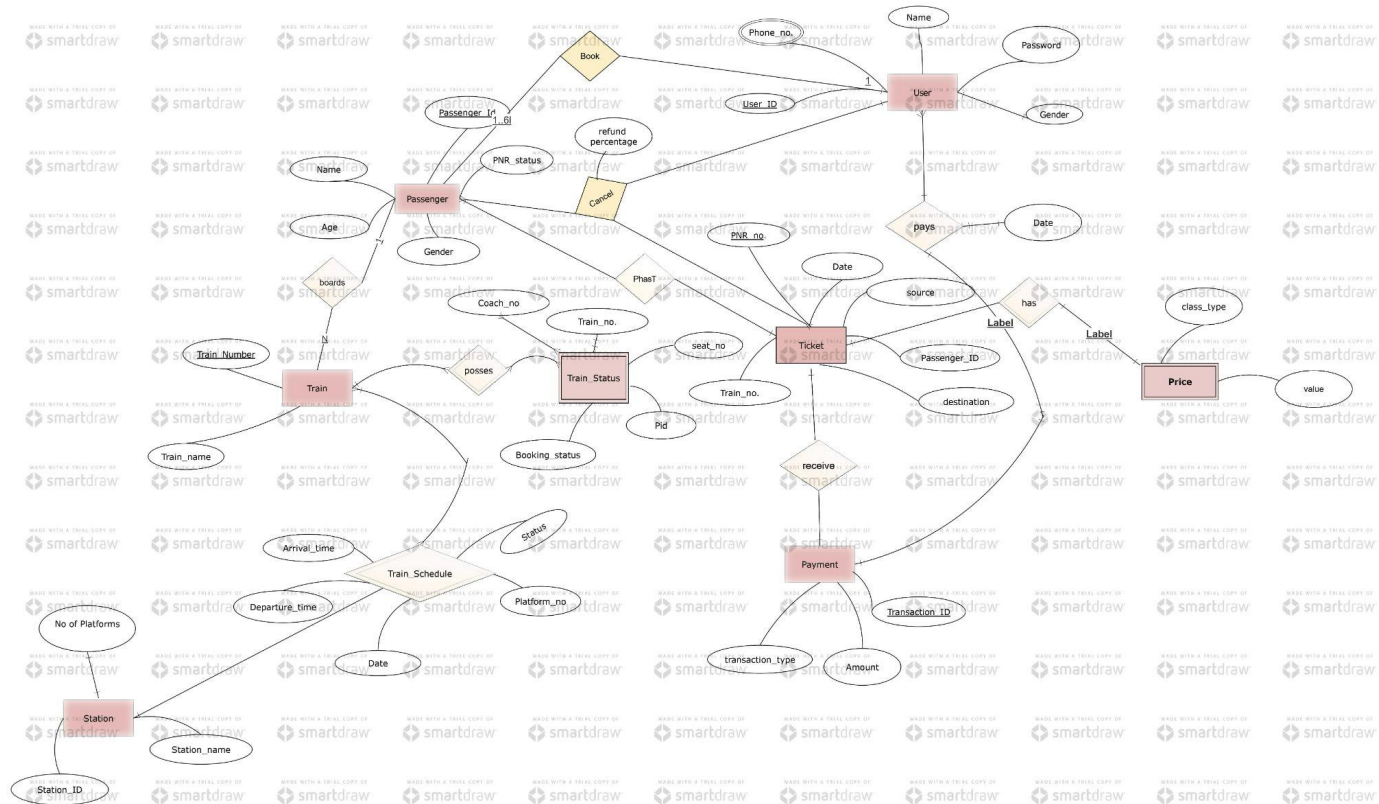
This system is reserved only for booking tickets, informations about the management of trains(kilometers and etc), drivers(how much is their salaries). In this system, the user can: see available trips from one location to another see all trips the user made in a month buy tickets the train system control, can track all information about the passengers as well as the payments.

# Stakeholders:

---



# Entity Relationship(ER) Diagram:



# Relation Schema:

---

## Relation Schema (**Bold** represents relations)

Train (train\_no., #train\_name, #station\_no., #p\_id)

Station (station\_no., station\_id, no.\_of\_platforms)

Train\_status (coach\_no., seat\_no., p\_id, booking\_status)

Ticket (#p\_id, #PRN\_no, Date, destination, source, train\_no., #transacton\_id)

Payment (transaction\_id, amount, transaction\_type)

Price (class\_type, value, p\_id)

Passenger (p\_id, name, age, gender)

User (u\_id, , name, password, gender, #p\_id, #transaction\_id)

**Train\_schedule** (arrival\_time, Date, status, platform\_no, dep\_time)

**Pays** (Date)

## Views and Grants:

View for passenger's booking details

```
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `railwaybooking`.`passenger_booking_details_1` AS
SELECT
    `p`.`Name` AS `name`,
    `p`.`Age` AS `age`,
    `p`.`Gender` AS `gender`,
    `t`.`Train_Number` AS `Train_number`,
    `t`.`Source` AS `source`,
    `t`.`Destination` AS `destination`,
    `t`.`Date` AS `date`,
    `ts`.`Coach_No` AS `coach_no`,
    `ts`.`Seat_No` AS `seat_no`
FROM
    ((`railwaybooking`.`passenger` `p`
    JOIN `railwaybooking`.`ticket` `t` ON (((`p`.`Passenger_ID` = UTL())
    AND (`p`.`Passenger_ID` = `t`.`Passenger_ID`))))
    JOIN `railwaybooking`.`train_status` `ts` ON (((`p`.`Passenger_ID` = UTL())
    AND (`ts`.`Pid` = `p`.`Passenger_ID`))));;
```



## View for showing train schedule

```
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `railwaybooking`.`train_schedule_view` AS
SELECT
    `railwaybooking`.`train_schedule`.`Station_ID` AS `station_id`,
    `railwaybooking`.`train_schedule`.`Platform_No` AS `Platform_No`,
    `railwaybooking`.`train_schedule`.`Arrival_Time` AS `Arrival_Time`,
    `railwaybooking`.`train_schedule`.`Departure_Time` AS `Departure_Time`,
    `railwaybooking`.`train_schedule`.`Status` AS `status`
FROM
    `railwaybooking`.`train_schedule`
WHERE
    (`railwaybooking`.`train_schedule`.`Train_number` = UTL())
ORDER BY `railwaybooking`.`train_schedule`.`Arrival_Time`;;
```

# TRIGGERS:

---

Trigger for checking age range.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `passenger_BEFORE_INSERT`  
BEFORE INSERT ON `passenger` FOR EACH ROW BEGIN  
    DECLARE msg varchar(128) default '';  
    IF NEW.age < 3 OR NEW.age > 120  
    THEN SET msg = "Age should be between 3 and 120";  
    END IF;  
  
    IF msg != ''  
    THEN SIGNAL SQLSTATE '45000'  
    SET MESSAGE_TEXT = msg;  
    END IF;  
END
```

---

Trigger for checking minimum price constraint for change ticket price.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `price_BEFORE_UPDATE` BEFORE UPDATE ON `price` FOR EACH ROW BEGIN
DECLARE msg varchar(128) default '';
    IF NEW.price < 100 THEN SET msg = "Price should be > 100";
    END IF;

    IF msg != '' THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
    END IF;
END
```

---

Trigger for checking if the same train already exists.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `train_BEFORE_INSERT` BEFORE INSERT ON `train` FOR EACH ROW BEGIN
    DECLARE msg varchar(128) default '';

    IF NEW.train_name in (select train_name from train )
    THEN SET msg = "Train already exists";
    END IF;

    IF msg != ''
    THEN SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = msg;
    END IF;
END
```

---

Trigger for checking if the station is present on the input train.( while changing platform number

```
CREATE DEFINER=`root`@`localhost` TRIGGER `train_schedule_BEFORE_UPDATE` BEFORE UPDATE ON `train_schedule` FOR EACH ROW BEGIN
    DECLARE msg varchar(128) default '';
    DECLARE tno int;
    declare sno int;
    declare temp int;

    select count(*) from train_schedule where train_number = NEW.train_number AND station_id = NEW.station_id into temp;
    IF temp = 0
    THEN SET msg = "Station is not present in the train's route";
    END IF;

    IF msg != '' THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
    END IF;

END
```

# Trigger for checking if the mobile number is already registered while signing up.

```
DROP TRIGGER IF EXISTS `railwaybooking`.`check_phone_registered`;  
  
DELIMITER $$  
USE `railwaybooking`$$  
CREATE DEFINER=`root`@`localhost` TRIGGER `check_phone_registered` BEFORE INSERT ON `user` FOR EACH ROW BEGIN  
    DECLARE message VARCHAR(128) DEFAULT '';  
    declare pnum varchar(10);  
  
    if NEW.phone_number in (select phone_number from user) then  
        set message = "Phone Number already registered";  
    end if;  
  
END;$$  
DELIMITER ;
```

# Grants

---

```
create user 'admin'@'127.0.0.1' identified by '1234';  
grant all on railwaybooking to 'admin'@'127.0.0.1';
```

```
create user 'user'@'127.0.0.1' identified by 'code';  
grant all on railwaybooking to 'user'@'127.0.0.1';
```

# Indexing:

---

```
ALTER TABLE `railwaybooking`.`train_status`  
ADD INDEX `tdate` (`Date` ASC) VISIBLE;  
;
```

```
ALTER TABLE `railwaybooking`.`train_schedule`  
ADD INDEX `tnumber` (`Train_number` ASC, `Station_ID` ASC) VISIBLE;  
;
```

```
ALTER TABLE `railwaybooking`.`ticket`  
ADD INDEX `tpnr` (`PNR_Number` ASC) VISIBLE;  
;
```

```
ALTER TABLE `railwaybooking`.`pays`  
ADD INDEX `tuser` (`User_ID` ASC) VISIBLE;  
;
```

```
ALTER TABLE `railwaybooking`.`book`  
ADD INDEX `buser` (`User_ID` ASC) VISIBLE;  
;
```



# SQL QUERIES

---

```
-- Train boarded by passenger with given pid.  
SELECT  
    train name  
FROM  
    train  
WHERE  
    train number IN (SELECT  
        train no  
    FROM  
        boards  
    WHERE  
        passenger id = $pid)  
LIMIT 1;
```

```
-- Train arrival time of passenger with given pid.
```

```
SELECT
    TS.Arrival Time
FROM
    train_schedule TS
WHERE
    (TS.Station ID IN (SELECT
        S.Station ID
    FROM
        station S
    WHERE
        S.Station name IN (SELECT
            T.source
        FROM
            ticket T
        WHERE
            T.Passenger ID = $pid))
    AND TS.Train number IN (SELECT
        T.Train Number
    FROM
        ticket T
    WHERE
        T.Passenger ID = $pid))
LIMIT 1;
```

```
-- First empty seat of a given coach and given train.
```

---

```
SELECT
    seat no
FROM
    train status
WHERE
    coach no = '$c'
    AND train no IN (SELECT
        train number
        FROM
            train
        WHERE
            train name = '$train')
    AND booking status = 0
    AND date = '$date'
LIMIT 1;
```

```
-- inserting values into ticket and boards
```

---

```
INSERT INTO ticket VALUES($newPNR, (SELECT train number FROM train WHERE train name = '$train'),  
'$orCity', '$dtCity', '$date', $newPassID);
```

```
INSERT INTO boards VALUES($newPassID, (SELECT train number FROM train WHERE train name = '$train'));
```

---

```
-- updating platform no of the given train with given station.
```

```
UPDATE train schedule
```

```
SET
```

```
    platform no = $newplat
```

```
WHERE
```

```
    train number IN (SELECT
```

```
        train number
```

```
        FROM
```

```
        train
```

```
        WHERE
```

```
            train name = '$sname')
```

```
    AND station id IN (SELECT
```

```
        station id
```

```
        FROM
```

```
        station
```

```
        WHERE
```

```
            station name = '$sname') LIMIT 1;
```

```
-- Renewing booked seats of all trains on a day.
```

```
DROP procedure IF EXISTS `railwaybooking`.`updateTrain_status date`;
```

---

```
;
```

```
DELIMITER $$
```

```
USE `railwaybooking`$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `updateTrain_status date`()
```

```
BEGIN
```

```
DECLARE cur date;
```

```
declare cur2 int;
```

```
declare cur3 date;
```

```
declare temp date;
```

```
declare temp1 date;
```

```
declare temp2 date;
```

```
declare temp3 date;
```

```
declare temp4 date;
```

```
declare temp5 int;
```

```
SELECT  
    MAX(date)  
FROM  
    train_status INTO cur;
```

---

```
SELECT  
    COUNT(train_no)  
FROM  
    train_status  
WHERE  
    date = CURDATE() INTO cur2;
```

```
SELECT  
    MIN(date)  
FROM  
    train_status INTO cur3;
```

```
if cur = curDate() then  
    update train_status set date = date add(date, interval 3 day) where date != curDate();  
UPDATE train_status  
SET  
    booking_status = 0,  
    pid = 0;
```

```
elseif cur2 = 0 then
    select min(date) from train_status into temp;
    UPDATE train_status
SET
    date = CURDATE()
WHERE
    date = temp;
SELECT
    MIN(date)
FROM
    train_status INTO temp1;
UPDATE train_status
SET
    date = DATE_ADD(CURDATE(), INTERVAL1 DAY)
WHERE
    date = temp1;
SELECT
    MIN(date)
FROM
    train_status INTO temp2;
UPDATE train_status
SET
    date = DATE_ADD(CURDATE(), INTERVAL2 DAY)
WHERE
    date = temp2;
    UPDATE train_status
SET
    booking_status = 0,
    pid = 0;
```



```
elseif cur3 = curdate() then  
    select count(train no) from train status into temp5;
```

---

```
else  
    select min(date) from train status into temp4;  
    UPDATE train status  
SET  
    date = DATE ADD(date, INTERVAL 3 DAY)  
WHERE  
    date = temp4;  
    UPDATE train status  
SET  
    booking status = 0,  
    pid = 0;  
end if;
```

```
END;$$
```

```
DELIMITER ;  
;
```

```
CALL updateTrain status date();
```

```
-- updating train status(whether departed or at staion or arriving).
```

```
DROP procedure IF EXISTS `railwaybooking`.`update trainSchedule status`;  
;
```

---

```
DELIMITER $;
```

```
USE `railwaybooking`$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `update trainSchedule status`()
```

```
BEGIN
```

```
    UPDATE train_schedule
```

```
SET
```

```
    status = 0
```

```
WHERE
```

```
    CURTIME() > Departure_Time and date = curDate() ;
```

```
    UPDATE train_schedule
```

```
SET
```

```
    status = 2
```

```
WHERE
```

```
    CURTIME() < Departure_Time and date = curDate() ;
```

```
UPDATE train schedule
SET
    status = 1
WHERE
    CURTIME() < Departure Time
    AND CURTIME() > arrival time and date = curDate();
```

```
UPDATE train schedule
SET
    Date = CURDATE()
WHERE
    Date < CURDATE();
END;$$
```

```
DELIMITER ;
;
```

```
CALL update trainSchedule status();
DROP procedure IF EXISTS `railwaybooking`.`updateTrain status`;
;
```

```
-- Booking seat in the given train and coach for the given passenger on a given date.
```

```
DELIMITER $$
```

```
USE `railwaybooking`$$
```

---

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `updateTrain status`(in pid int, in tno int, in cno  
varchar(1), in sno int, in dat date)
```

```
BEGIN
```

```
    UPDATE train status T
```

```
SET
```

```
    Booking status = 1,
```

```
    T.Pid = pid
```

```
WHERE
```

```
    T.Booking status = 0
```

```
    AND T.Train no = tno
```

```
    AND T.coach no = cno
```

```
    And T.seat no = sno
```

```
    AND T.date = dat;
```

```
END$$
```

```
DELIMITER ;
```

```
;
```

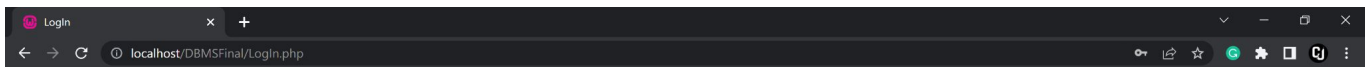
---

```
-- Deleting cancelled ticket
```

```
DELETE FROM ticket WHERE pnr number = $cancelPnr;
```

```
-- Selecting maximum statio id for adding a new station.
```

```
SELECT MAX(station id) FROM station;
```



**LOG IN**

User ID:

\_\_\_\_\_

Password:

\_\_\_\_\_

Submit

Don't have an account? [Sign Up](#)





## SIGN UP

Full Name:

Gender:

- ☐ Male  
☐ Female  
☐ Other

Phone No:

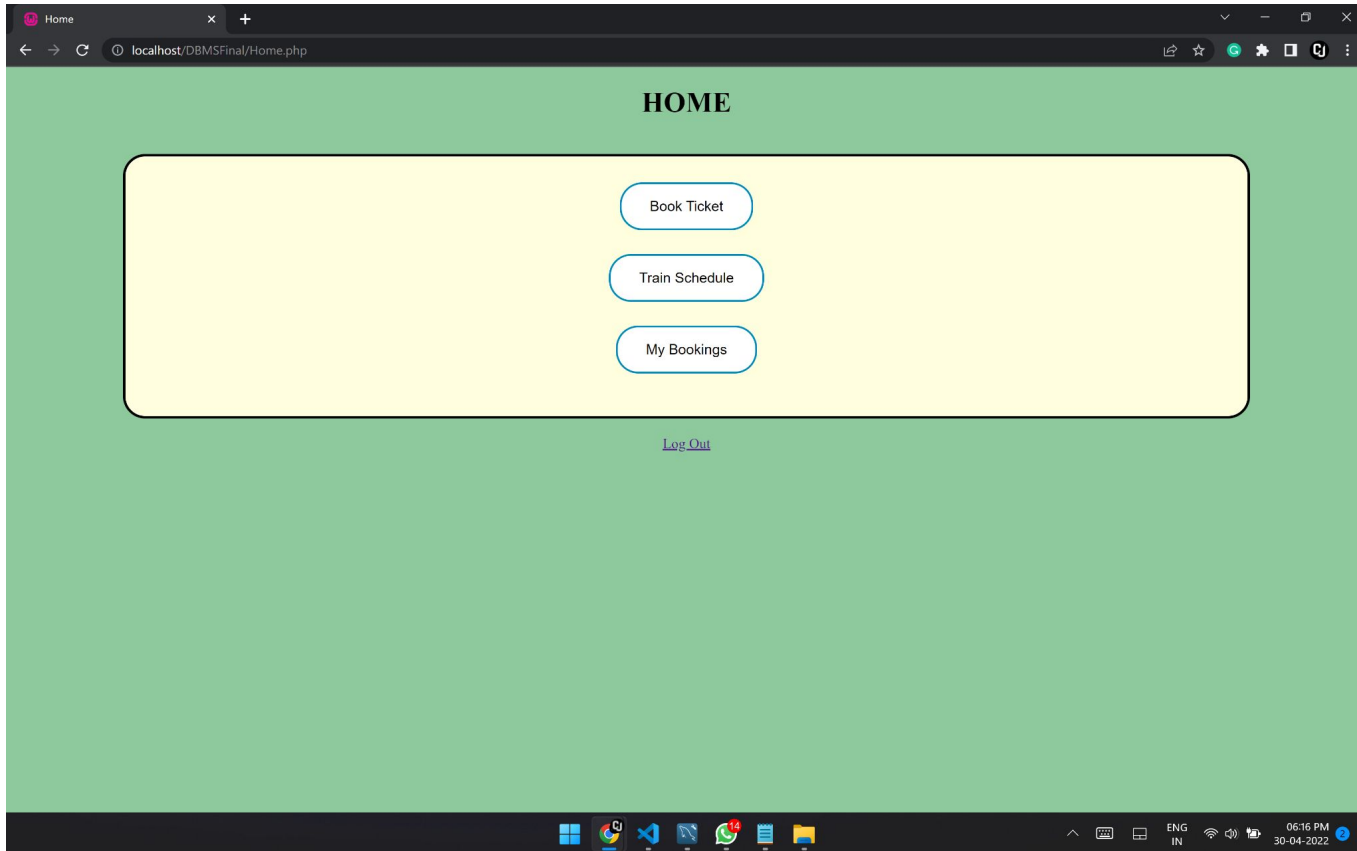
Password:

Confirm Password:

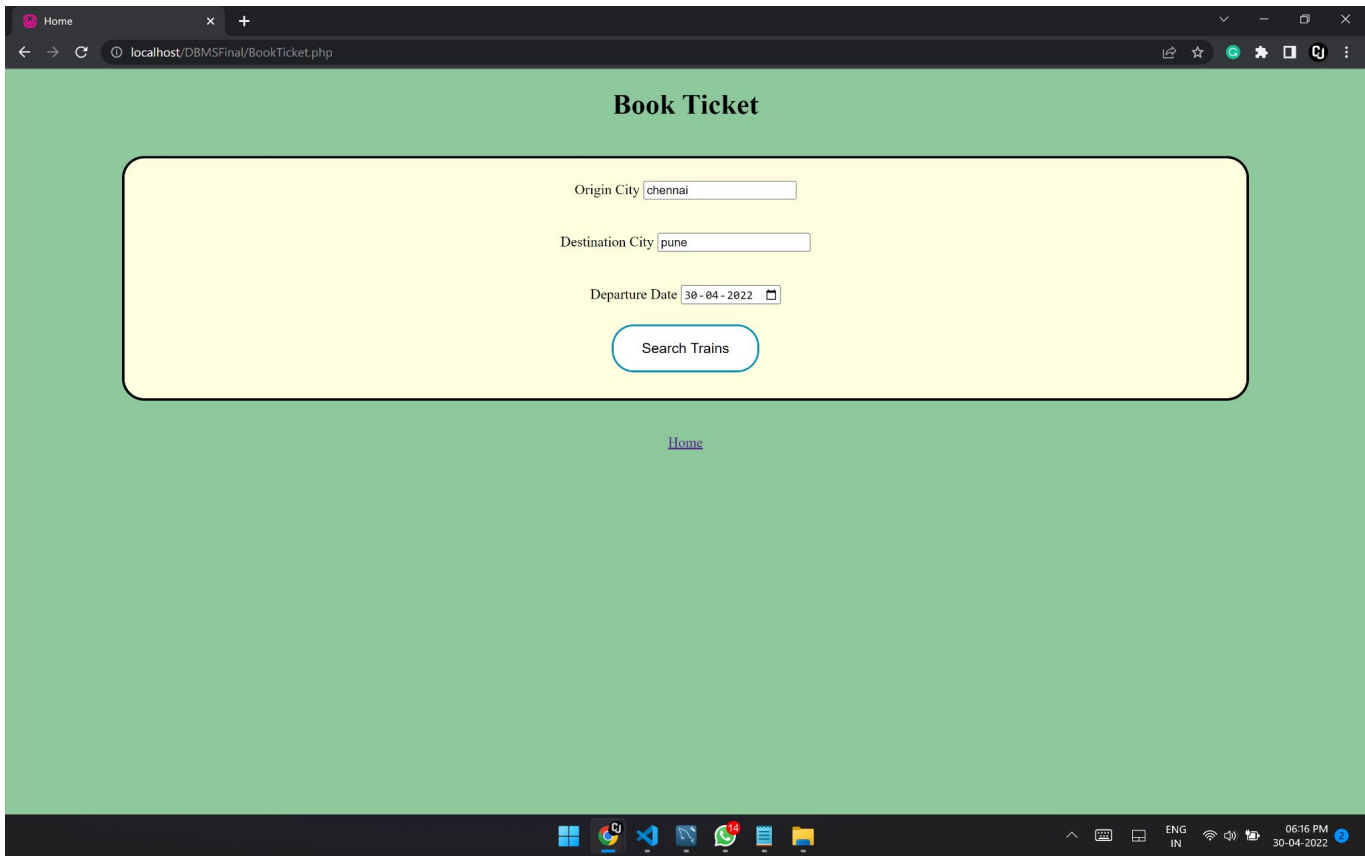
Already have an account? [Log In](#)

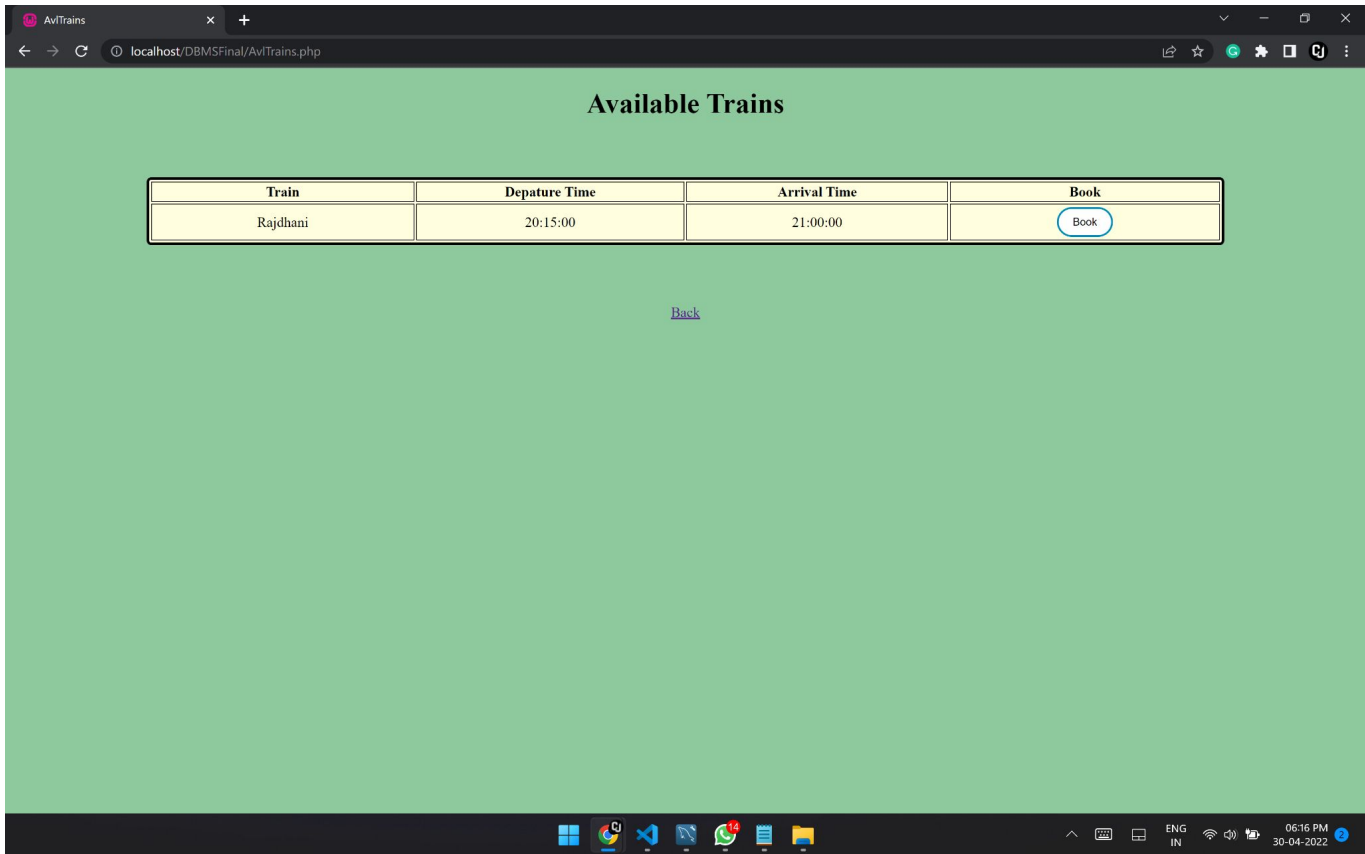


# User Interface:









Book Seat

localhost/DBMSFinal/BookSeat.php

Book Seat

Rajdhani

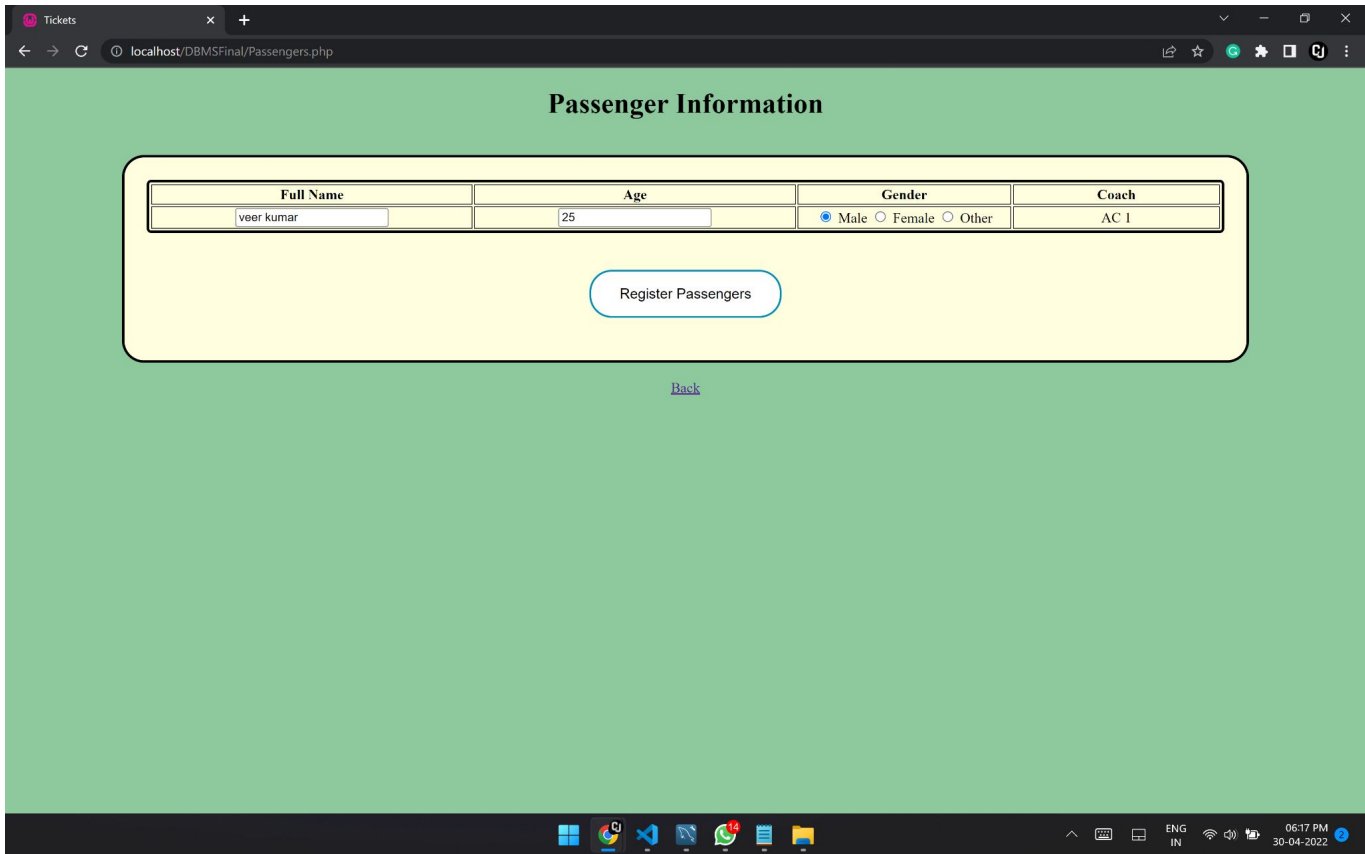
Coach	Seats Available	# Book Seats	Price (in Rs.)
AC 1	13	<input type="text" value="1"/>	2000
AC 2	21	<input type="text" value="0"/>	1500
AC 3	21	<input type="text" value="0"/>	1200
Sleeper	41	<input type="text" value="0"/>	800
Genaral	31	<input type="text" value="0"/>	201

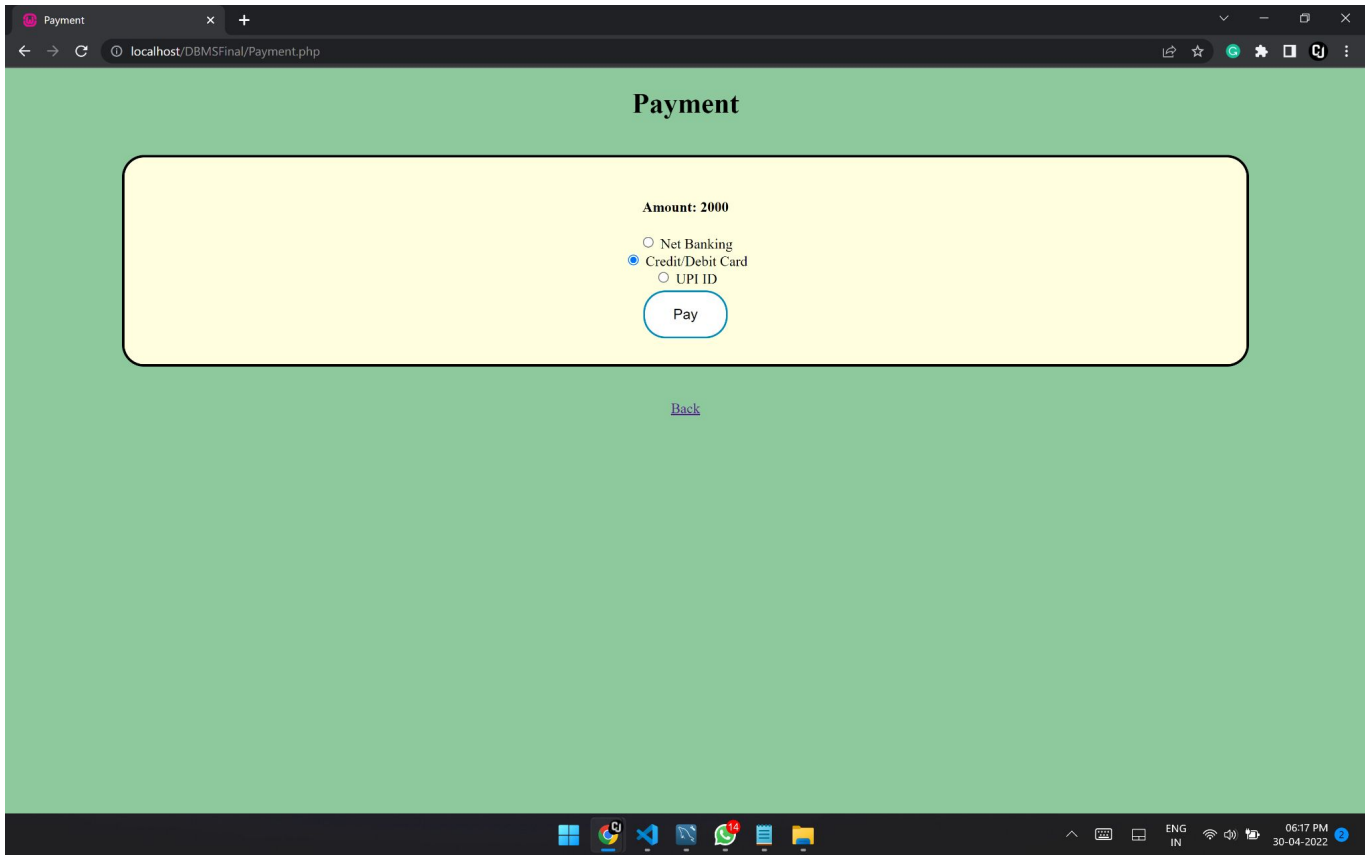
Book Seats

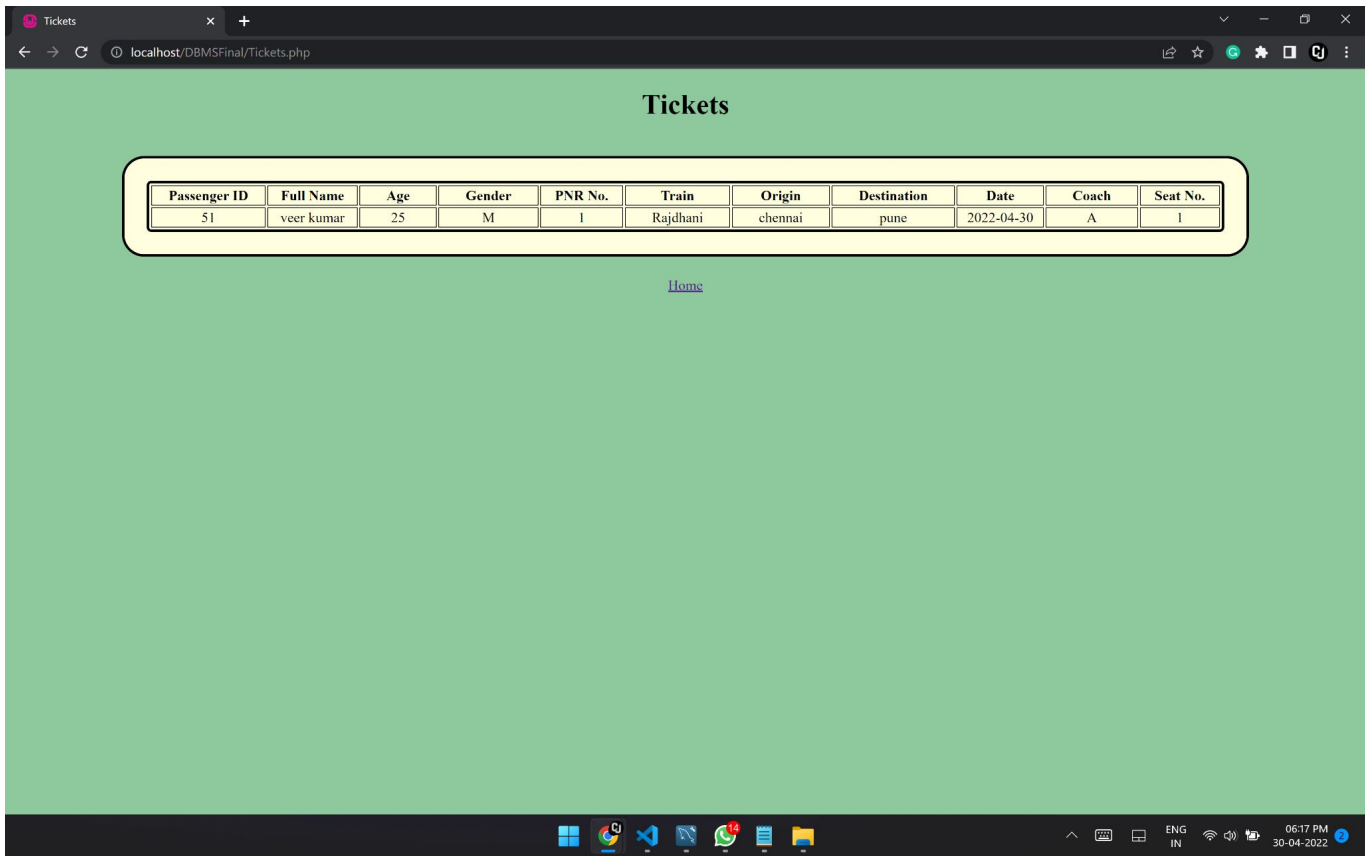
[Back](#)

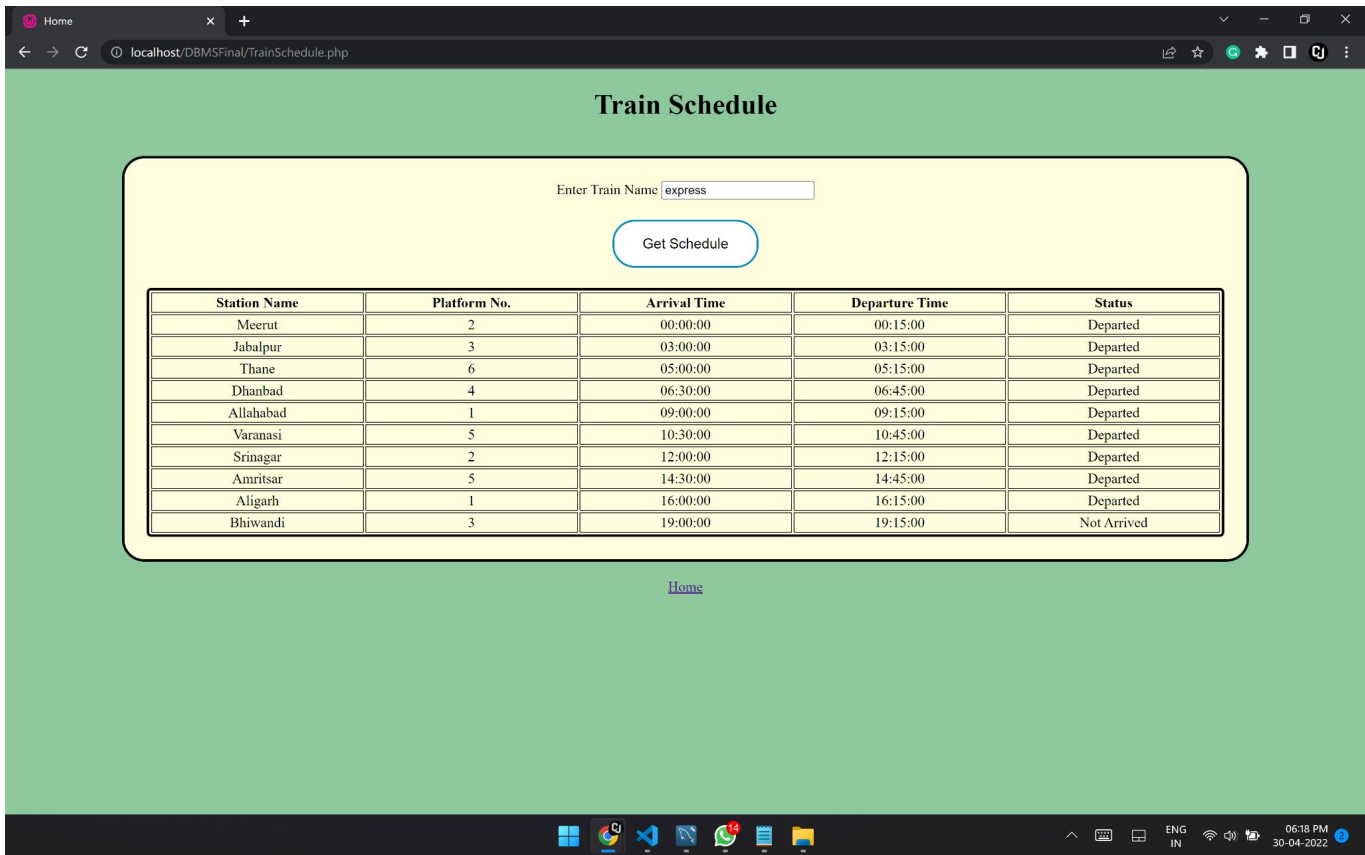
Windows Taskbar

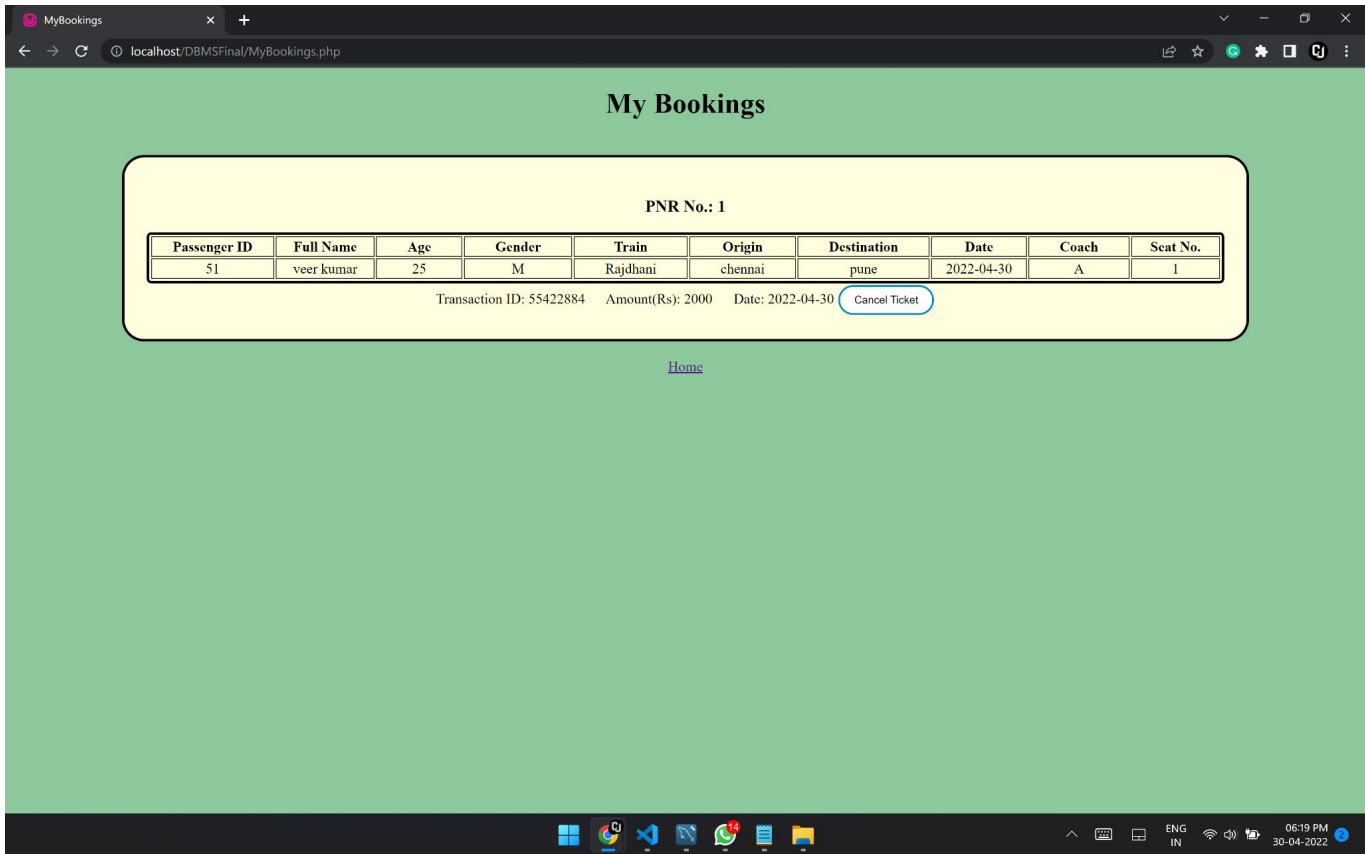
06:17 PM 30-04-2022



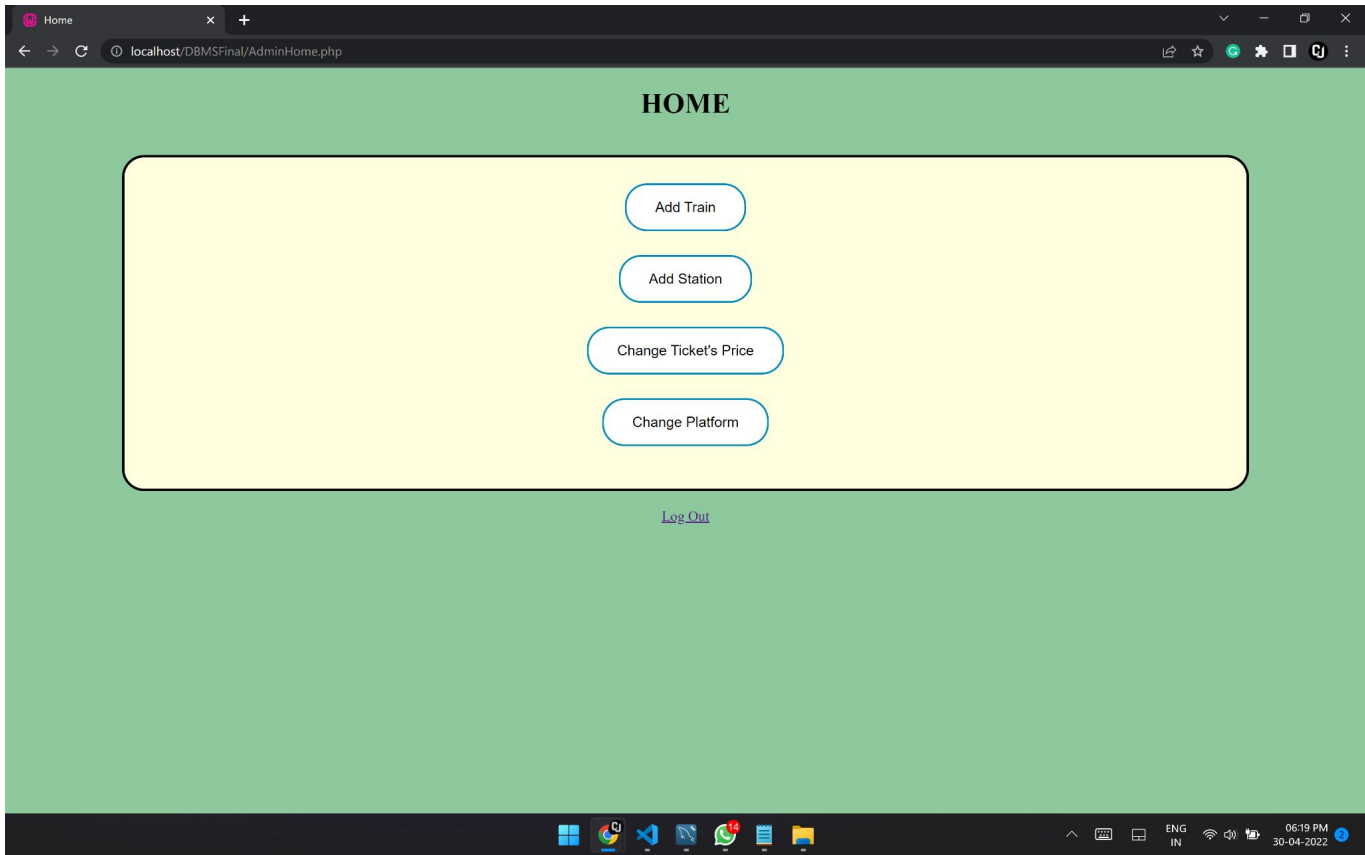


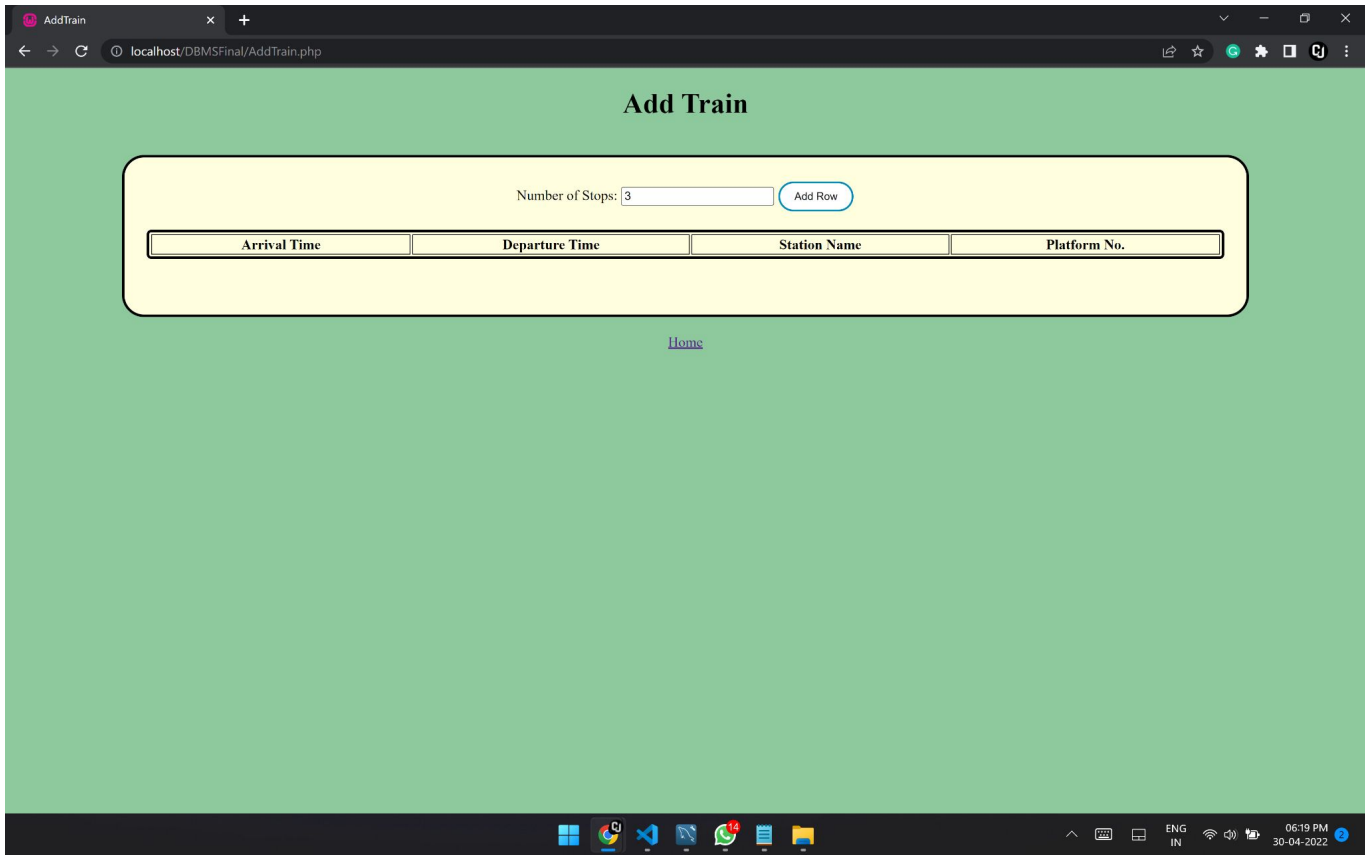


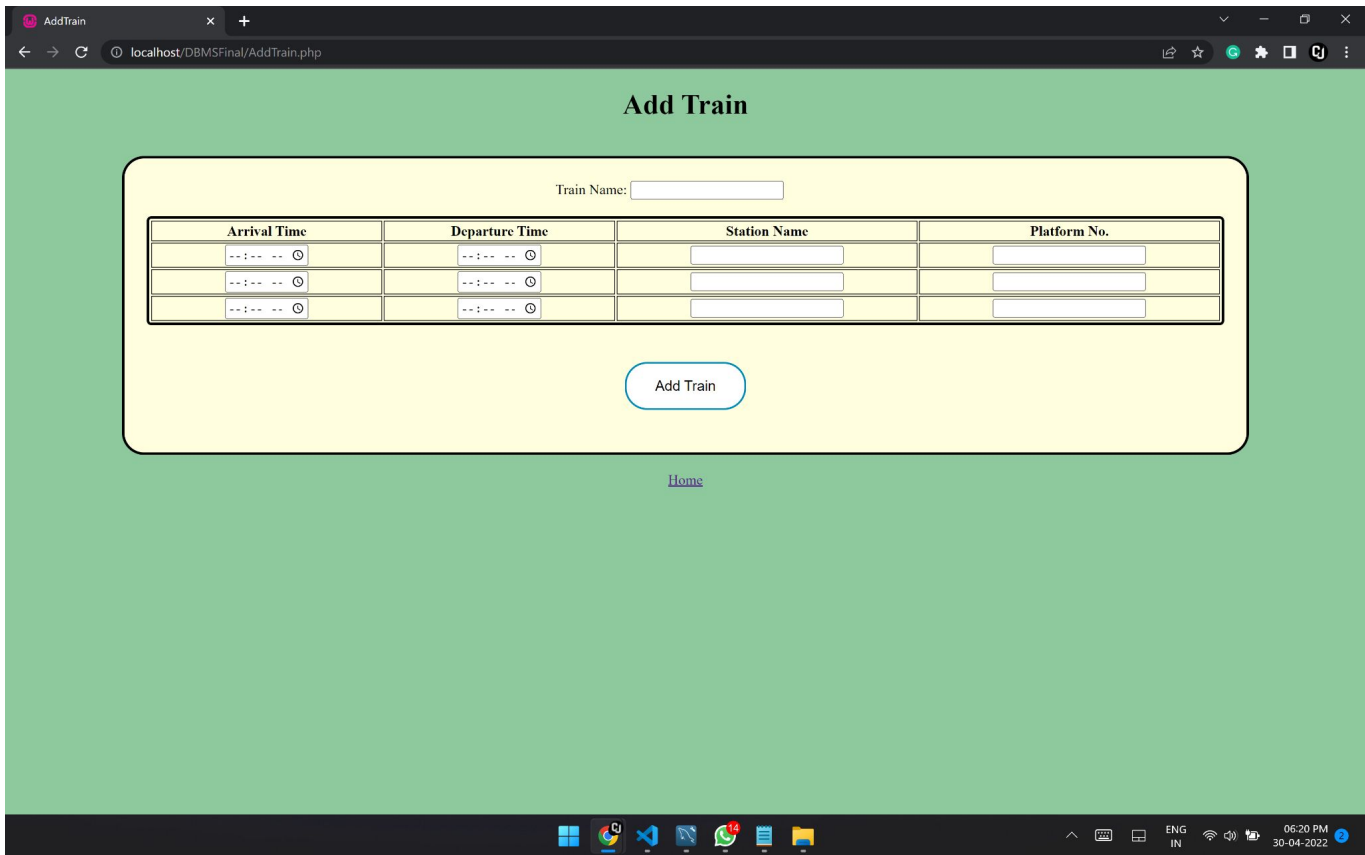


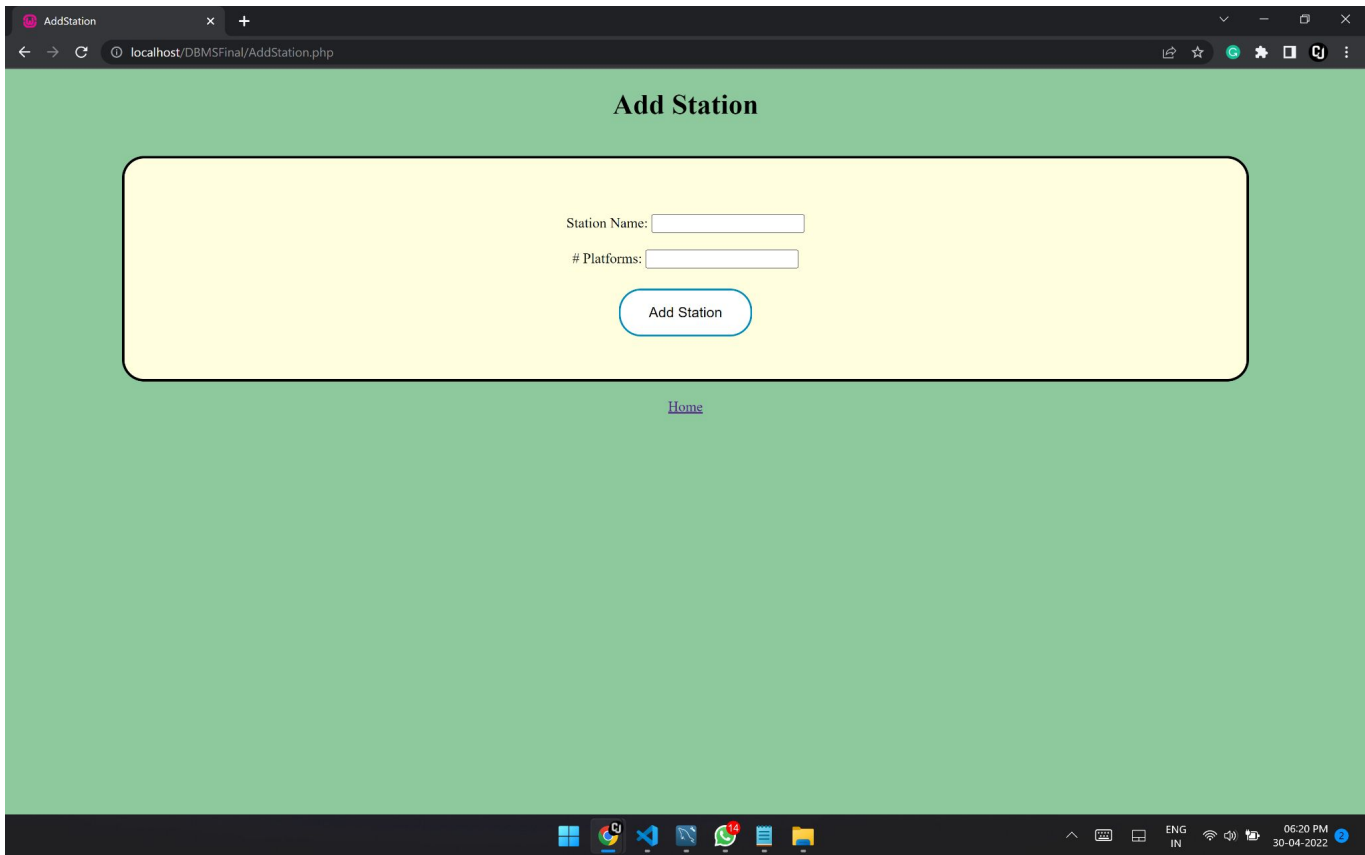


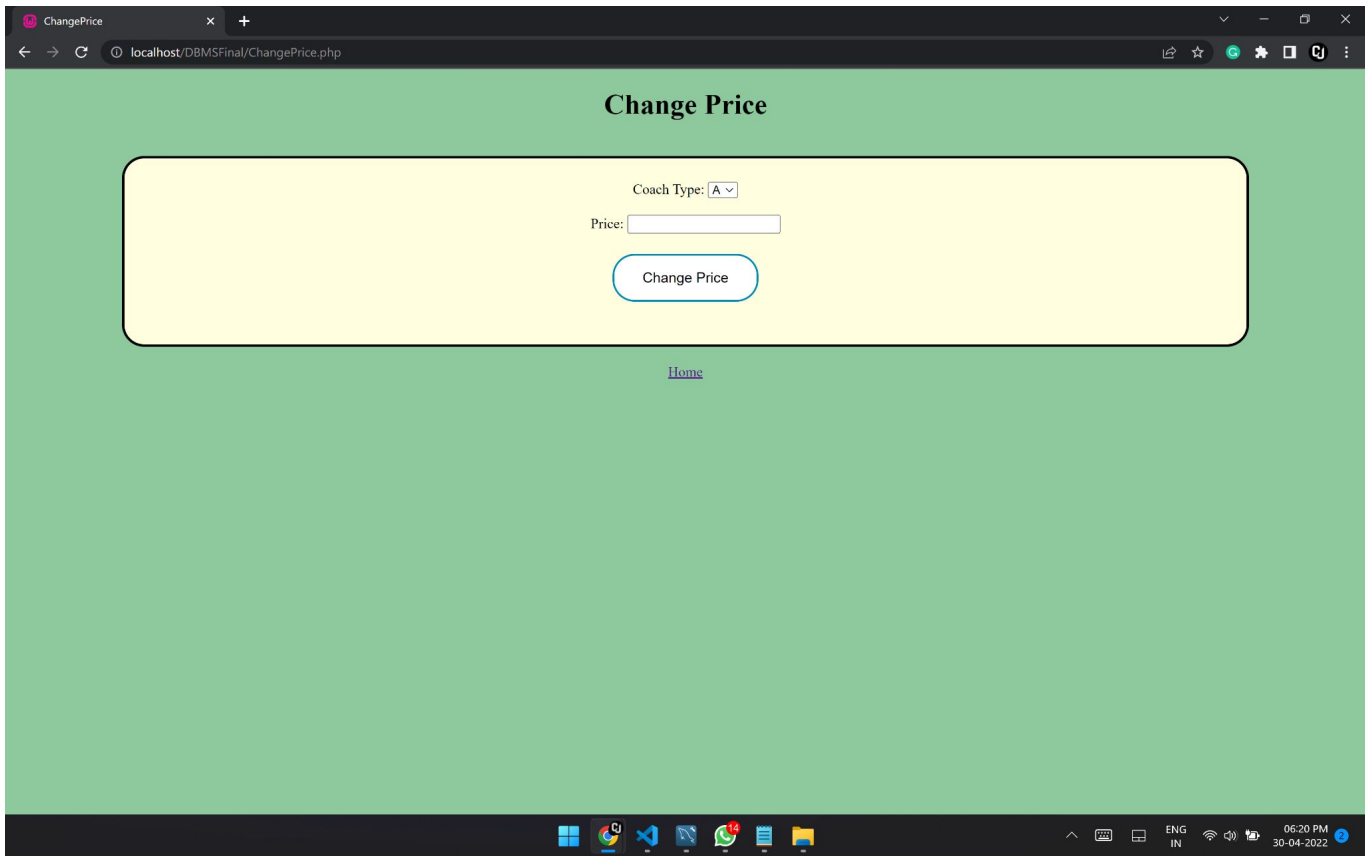


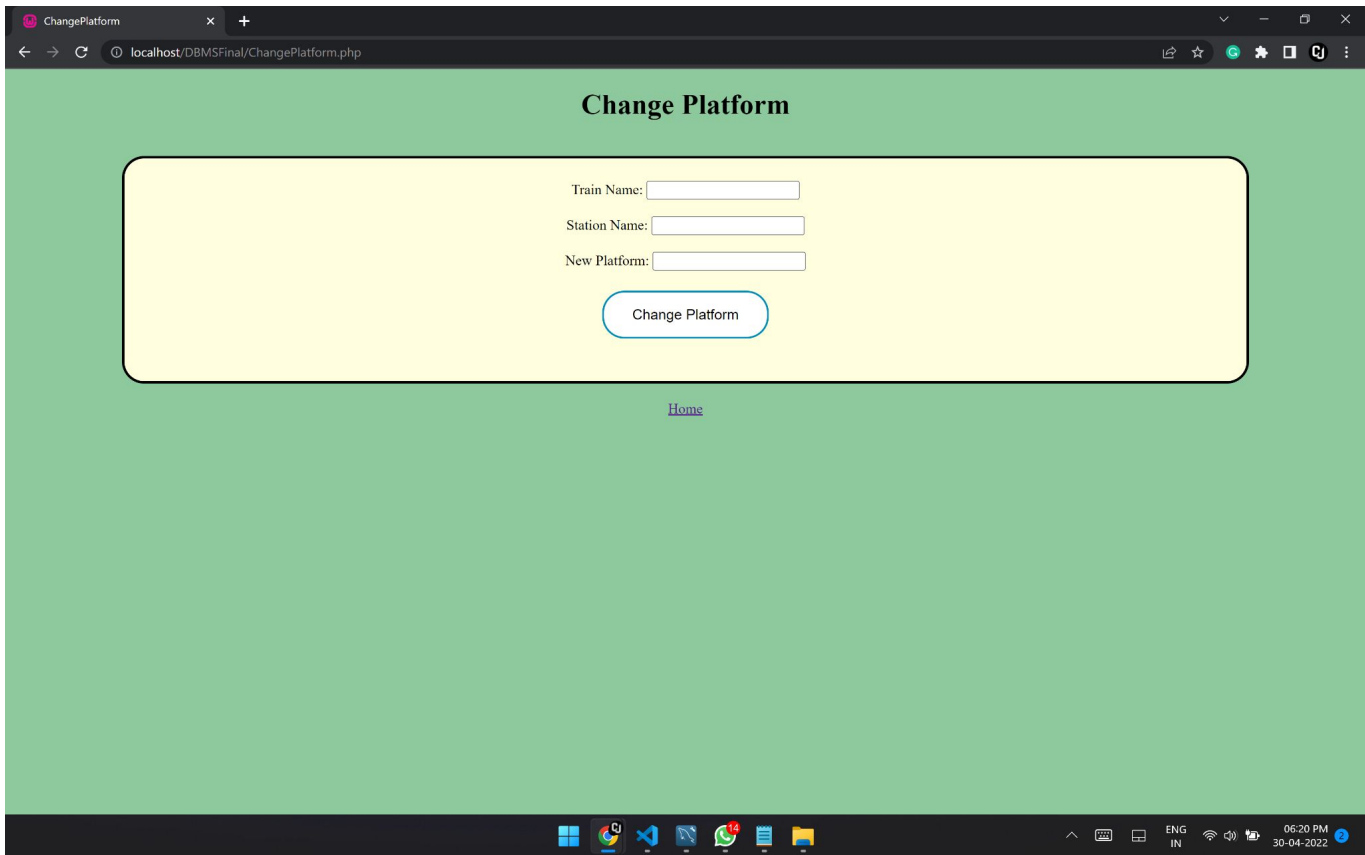












# Embedded Query

---

```
$i = 0;
while ($i < count($a)) {
    $query = "SELECT
        Departure_Time
    FROM
        train_schedule
    WHERE
        Train_Number IN (SELECT
            Train_Number
        FROM
            Train T
        WHERE
            T.Train_Name = '$a[$i]')
        AND Station_ID IN (SELECT
            Station_ID
        FROM
            station
        WHERE
            Station_Name = '$orCity')";
    $res1 = mysqli_query($conn, $query);
    while ($row = mysqli_fetch_row($res1)) {
        array_push($dtime, $row[0]);
    }
    mysqli_free_result($res1);
    $i+=1;
}
```

```
mysqli_query($conn, "INSERT INTO passenger VALUES('$n', $newPassID, $a, '$g')");
mysqli_query($conn, "INSERT INTO ticket VALUES($newPNR, (SELECT train_number FROM train WHERE train_name = '$train'), '$orCity', '$trainNo', '$date')");
mysqli_query($conn, "INSERT INTO boards VALUES($newPassID, (SELECT train_number FROM train WHERE train_name = '$train'))");
$trainNo = mysqli_fetch_row(mysqli_query($conn, "SELECT train_number FROM train WHERE train_name = '$train'))[0];

mysqli_query($conn, "CALL updateTrain_status($newPassID, $trainNo, '$c', $seat, '$date')");
// mysqli_query($conn, "CALL updateTrain_status($newPassID, $trainNo, '$ct', '$seat')");

$amount += mysqli_fetch_row(mysqli_query($conn, "SELECT price from price where coach = '$c'"))[0];
mysqli_query($conn, "INSERT INTO phast VALUES($newPassID, $newPNR)");
mysqli_query($conn, "INSERT INTO book VALUES($uid, $newPassID)");
$newPassID++;
// $newPNR++;
mysqli_query($conn, "INSERT INTO receive values($newPNR, $tranID)");
```



```
$tno = mysqli_fetch_row(mysqli_query($conn, "SELECT train_number from train where train_name = '$trainName' limit 1"))[0];
$query = "SELECT * from (select @utl:=$tno) t, train_schedule_view;";

$res = mysqli_query($conn, $query);
$flag = 0;
while ($row = mysqli_fetch_row($res)) {
    $r = $row[1];
    $temp = mysqli_fetch_row(mysqli_query($conn, "SELECT station_name from station where station_id = $r limit 1;"))[0];
    array_push($sName, $temp);
    array_push($pNo, $row[2]);
    array_push($aTime, $row[3]);
    array_push($dTime, $row[4]);
    if ($row[5] == 0) {
        array_push($status, 'Departed');
    }
    else if ($row[5] == 1) {
        array_push($status, 'At Station');
    }
    else if ($row[5] == 2) {
        array_push($status, 'Not Arrived');
    }
    else {
        array_push($status, '---');
    }
    $flag = 1;
}
if ($flag == 0) {
    echo '<script type="text/javascript"> alert("The Train dosen\'t exist"); </script>';
}
```

```

for ($j = 1; $j <= $_SESSION['i']; $j++) {
    if (isset($_POST['cancel'.$j])) {
        unset($_POST['cancel'.$j]);
        $cancel_pnr = $_SESSION['cancelPnr'];
        $cancelPnr = $cancel_pnr[$j-1];
        $uid = $_SESSION['uid'];
        // add in cancel
        mysqli_query($conn, "INSERT INTO cancel VALUES ($uid, $cancelPnr, 50)");
        // get pid
        $pids = array();
        $res = mysqli_query($conn, "SELECT passenger_id FROM ticket WHERE pnr_number = $cancelPnr");
        while ($row = mysqli_fetch_row($res)) {
            array_push($pids, $row[0]);
        }
        // update train status
        for ($k = 0; $k < count($pids); $k++) {
            $p = $pids[$k];
            mysqli_query($conn, "UPDATE train_status SET booking_status = 0, pid = 0 WHERE pid = $p");
        }
        $tranID = mysqli_fetch_row(mysqli_query($conn, "SELECT transaction_id FROM receive WHERE pnr_no = $cancelPnr LIMIT 1"))[0];
        // del payment
        mysqli_query($conn, "DELETE FROM payment WHERE transaction_id = $tranID");
        // del receive
        mysqli_query($conn, "DELETE FROM receive WHERE pnr_no = $cancelPnr LIMIT 1");
        // del book
        // del passanger
        for ($k = 0; $k < count($pids); $k++) {
            $p = $pids[$k];
            mysqli_query($conn, "DELETE FROM passenger WHERE passenger_id = $p");
            mysqli_query($conn, "DELETE FROM book WHERE passenger_id = $p");
            mysqli_query($conn, "DELETE FROM boards WHERE passenger_id = $p");
        }
        // del phat
        mysqli_query($conn, "DELETE FROM phast WHERE pnr_no = $cancelPnr");
        // del ticket
        mysqli_query($conn, "DELETE FROM ticket WHERE pnr_number = $cancelPnr");

        echo "<script> alert('Ticket Cancelled'); </script>";
    }
}

```

---