A PROJECT REPORT
ON

Online Railway Reservation System

Submitted in partial fulfillment of the
Requirements for the degree of

B.Tech in Information Technology

by

Parama Sarbajna
Sanwar Hussain

Department of Information Technology
Narula Institute of Technology
An Autonomous Institute under
Maulana Abul Kalam Azad University of Technology
West Bengal University of Technology

2017-2018

Certificate of Approval

This is to certify that this dissertation entitled **Online Railway Reservation System** submitted in partial fulfillment of the requirements for the degree of **B.Tech** in **Information Technology** from **Narula Institute of Technology** under **West Bengal University of Technology** which is the result of the bonafied. Research work carried out by **Parama Sarbajna** and **Sanwar Hussain**.

It is understood that by this approval the undersigned do not necessarily endorse any of the statements made or opinion expressed therein but approves it only for the purpose for which it is submitted.

[Prof. (Dr.) Bidyut K Medya]
Head, Dept. of IT

[Mr. Soumen Ghosh]
Assistant Professor, Dept. of IT
Project Supervisor

DECLARATION

We, the under signed, do hereby declare that the project report submitted to the **Narula Institute of Technology** in partial fulfillment of the requirements for the degree of **B.Tech** in **Information Technology** entitled **Online Railway Reservation System**, is an original piece of research work carried out by us under the guidance and supervision of **Mr. Soumen Ghosh, Dept. of IT**

We further declare that the information has been collected from genuine & authentic source and we have not submitted this project report elsewhere.

Name	**University Reg. Nos.**	**Signature**
1. Parama Sarbajna	**141270110149**	
2. Sanwar Hussain	**141270110155**	

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Prof. (Dr.) Bidyut K Medya, for providing us an opportunity to do the project work in Narula Institute of Technology and giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

We owe our deepest gratitude to our project guide Mr. Soumen Ghosh, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Information Technology department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in laboratory for their timely support.

CONTENTS

Page No.

1. Introduction to project

Indian Railways (IR) is India's national railway system operated by the Ministry of Railways. It manages the fourth-largest railway network in the world by size, with 121,407 kilometers (75,439 mi) of total track over a 67,368-kilometre (41,861 mi) route. Forty nine percent of the routes are electrified with 25 KV AC electric traction while thirty three percent of them are double or multi-tracked.

IR runs more than 13,000 passenger trains daily, on both long-distance and suburban routes, from 7,349 stations across India. The trains have a five-digit numbering system. Mail or express trains, the most common types, run at an average speed of 50.6 kilometers per hour (31.4 mph).In the freight segment, IR runs more than 9,200 trains daily. The average speed of freight trains is around 24 kilometers per hour (15 mph).

As of March 2017, IR's rolling stock consisted of 277,987 freight wagons, 70,937 passenger coaches and 11,452 locomotives.IR owns locomotive and coach-production facilities at several locations in India

The world's eighth-largest employer, it had 1.308 million employees as of March 2017.

In the year ending March 2018, IR is projected to carry 8.26 billion passengers and transport 1.16 billion tons of freight. In the fiscal year 2017-18, IR is projected to have earnings of ₹1.874 trillion (US$29 billion), consisting of ₹1.175 trillion (US$18 billion) in freight revenue and ₹501.25 billion (US$7.7 billion) in passenger revenue, with an operating ratio of 96.0 percent.

This system is basically concerned with the reservation and cancellation of railway tickets to the passengers. The Need of this system arose because as is the known fact that India has the largest railway network in the whole of the world and to handle it manually is quite a tough job. By computerizing it, we will be able to overcome many of its limitations and will be able to make it more efficient. The handling of data and records for such a vast system is a very complex task if done manually but it can be made much easier if the system is computerized.

The basic functions being performed by our system are:

1. Reservation management

2. Fare management

3. Timetable management

These functions will be handled with the help of following sub functions: -

1. It reserves and cancels seats for the passenger.
2. It contains information about the trains.
3. It contains information about the passenger.
4. It contains the details of reservation fees.
5. It makes entries for confirmed, waiting tickets.

2. Objective of the project:

- The objective of railway reservation system is to give structural design to railway system.
- The project provides functionality and flexibility to railway system such that one can operate that system easily and efficiently.
- This project also provides a complete set of solutions for some common and specific areas of work in the railways.

3. SYSTEM REQUIREMENT SPECIFICATION

Product Functions
This software is that it is relatively simple to use. It Encompasses:

● Search: This function allows booking user to search for train that are available between the two travel station, namely the Departure station and the Arrival station as desired by the traveler. The system initially prompts the user for the departure and arrival station. It then displays the list of train available with different trains between designated destinations.

● Selection: This function allows a particular train to be selected from the displayed list. All the details of the train are shown:
1. Train Number
2. Place of Departure
3. Pace of Arrival
4. Arrival Time
5. Departure Time

● Traveler Information: It asks for the details of all passengers supposed to travel including name, age and gender.

● Cancellation: The system also allows the passenger to cancel an existing reservation. This function registers the information regarding a passenger who has requested for a cancellation of his/her attack.

1.1 User Classes and Characteristics

There are levels:
1. Database
2. User

Database: The databases include user database which has the details of the user logged in, train database that has the train schedule and the booking database that has details of trains and seat booked by the passenger.

Customer: They are a vital part of the system. They have access to book a ticket, search train, cancel a ticket, and view booked tickets using PNR numbers generated.

1.2 Operating Environment

- Distributed database
- Client/Server system
- Operating system: Windows and Linux
- MYSQL Database
- Language: PHP

1.3 Design and Implementation Constraints

The constraints are as follows:

- It allows only single user to login at a time.
- Information regarding cancellation and delay of trains are not displayed.
- Software is dependent on access to internet.
- The system shall be a web based application.
- The system must be user friendly.
- The development environment shall be Windows 8 or higher.
- The language used in the system must be easily understandable to the public.
- The computer must have sufficient memory.

1.4 User Documentation

The manual on how to use the website will be provided to the user. The user will have a manual which includes instructions on how to use the website.

.

1.5 Assumptions and Dependencies

- It is assumed that user will follow the proper instruction as indicated by the system.
- It is assumed that the users will possess decent internet connectivity.

1.6 External Interface Requirements

1.6.1 User Interfaces

- The frontend will be developed using several languages like html, css, JavaScript, php, bootstrap etc.
- Screen Format: The introductory screen will be the first to be displayed which will allow the users to enter the login details and check the availability of the ticket or cancel a ticket.

1.6.2 Hardware Interfaces

The system must basically support certain input and output devices. Their descriptions are as follows.

1.6.3 Software Interfaces

- Any Linux Based or Windows Based Operating System
- SQL Database

1.6.4 Communications Interfaces

- The system requires an HTTP to communicate with the server. The system can be configured to be accessed via any available port.
- The web based UI is the only means of communication between the user and the system.
- A proper internet connection is recommended.

2. System Features

2.1 Login Requirement

2.1.1 Description and Priority

It provides member authentication.

2.1.2 Stimulus/Response Sequences

The member is directed to main page on successful login. The input is verified by checking if the member already exists in the database. The correct input will result in the next page i.e. the analysis page being loaded. If the input is incorrect then an error message will be displayed.

2.2 Registration Form Requirement

2.2.1 Description and Priority

It provides registration of a non member. The user is asked to enter details like name, birthdate, address, contact number, email id and password.

2.2.2 Stimulus/Response Sequences

The input is validated using client side validation. The client side validation will include checks for missing information in the required fields and other text fields like email and phone numbers will be checked for validity. The appropriate error messages are displayed if the input is not acceptable

2.3 Passengers Information Management

2.3.1 Description and Priority

The system will maintain record of train no, allotted seat number and the date of travel. The system will allow creation/modification/deletion of new or existing passenger.

2.3.2 Stimulus/Response Sequences

User has to be logged in before any ticket details, cancellation details can be seen.

2.4 Trains Information Management

2.4.1 Description and Priority

The system will maintain information about the train name, train number. Number of seats in each train is fixed.

2.4.2 Stimulus/Response Sequences

Ticket information will be present in the system before it can be issued.

2.5 Issued Tickets Management

2.5.1 Description and Priority

The system will maintain information about seats that are issued. Corresponding passenger Details and date of travel.

2.5.2 Stimulus/Response Sequences

Passenger information and seat information must be entered before a seat can be allotted.
The issue details must be present before the seat is cancelled.

2.6 Cancel Ticket Information Management

2.6.1 Description and Priority

If a user wishes to cancel a ticket the user can do so.

2.6.2 Stimulus/Response Sequences

The entry will be removed from the database.

3. Other Nonfunctional Requirements

3.1 Performance Requirements

- The database should be scalable; it must have the capacity to hold large number of users in future.
- The number of connections to the system should not slow down the application to a large degree.
- The data for the analysis will be obtained from the database of users, so the response time for a query from the client side to the database side should not be more than 5 seconds.
- Error handling should be implemented and the application should be able to handle all runtime errors.
- The application should be flexible for future enhancements, for example, the addition of a few more additional features.

3.2 Safety Requirements

None

3.3 Security Requirements

- Users shall be required to log in to the RRS for their own reservation information and modification with email address and password.
- The system shall permit only authorized members who are in the authorized database

3.4 Software Quality Attributes

3.4.1 Usability

The website design shall allow deployment on both Windows and UNIX (Linux) servers.

3.4.2 Robustness

The system design shall include recovery scenarios allowing the ability to restore a state no older than one business day old.

3.4.3 Correctness

It should satisfy the normal regular HMS operations to fulfil end user objectives.

3.4.4 Efficiency

Resources should be implemented to achieve the particular task efficiency without hassle.

3.4.5 Flexibility

We should be able to add new features and handle them conveniently.

3.4.6 Integrity

System should focus on securing customer information and avoid data loss as much as possible.

3.4.7 Portability

System should run in any Windows or Linux system.

3.4.8 Maintainability

System should be maintainable.

3.5 Business Rules

None

3.6. Other Requirements

When the system is completely developed and submitted to the client, few sessions will be required to make the users of the system understand the functionality and adapt to the system. After these sessions, it is required that a member from the development team should spend some time in the system background for an agreed time period. That time period will be used to identify new bugs.

4. HARDWARE SPECIFICATIONS:

- Processor : Dual core i3 or equivalent or greater
- RAM : 1GB or Greater
- Hard disk : 80 GB or Greater
- Keyboard & Mouse
- Monitor : Color (For Best Result)
- Printer.

5. SOFTWARE REQUIREMENTS

- Operating System: Windows 7/8/8.1/10, Ubuntu or any other linux distro, Mac OS.
- Tools: Any browser preferably Chrome or Mozilla Firefox

6. FEASIBILITY STUDY

The feasibility study of the proposed development of the project may be analyses in three ways:

[1] Technical feasibility

[2] Economic feasibility

[3] Operational feasibility

1. Technical Feasibility: Technical Feasibility means analysis whether the system proposed for the development can be computerized or not. There is hardly any problem in the real world which cannot be computerized. In fact the hardware begging marketwise far too sophisticated for business application. This has change the emphasis in technical feasibility from whether to how to computerize these calls for a look on the software and computer expert rather than the hardware. The organization then need to know if there is people within the organization or without who can computerize the proposed project.

2. Economic Feasibility: It is basically cost profit analysis of proposed system development and is the decision whether to undertake or not to be undertake the development. It is relatively easier to measure cost and benefit which are usually intangible and far hard to quantify. The cost may be the composed of measure component which can then be further split as below—

(a) Startup Cost:

1. Salaries of system analyst, programmer during development stage.

2. Cost of Conversion, Preparation of data and system manual and other supportive document.

3. Cost of new additional Hardware if any.

4. Cost of training employees (if required)

(b) Operating Cost:

1. Hardware / Software rental or depreciation charge.
2. Salaries of maintenance staff.
3. Salaries of the operating staff.
4. Cost of input data preparation.
5. Installation and maintenance cost.
6. Overload cost charges of bus.

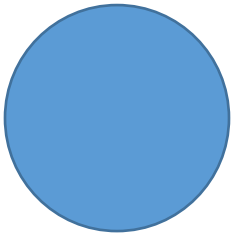
3. Operational Feasibility: This focuses on the willingness and ability of the management, employees, customers and suppliers of the organization to operate, use and support the proposed system.

DATA FLOW DIAGRAM:

The Data Flow Diagrams are pictorial or graphical representation of the outline of the system study. The data flow diagram covers all the processes and data storage area which takes place during any transaction in the system. The data flow diagrams are functionally divided into context level, Zero level, First level and Second level data flow diagrams.

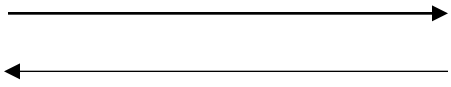
Symbols used in DFDs:

(1) Process: Here flow of data is transformed. E.g. Issue, Deposit, etc.



(2) External Entity: A source or destination of data which is external to the system. E.g. Student, Teacher etc.

(3) A data flow: It is packet of data. It may be in the form of document, letter etc.

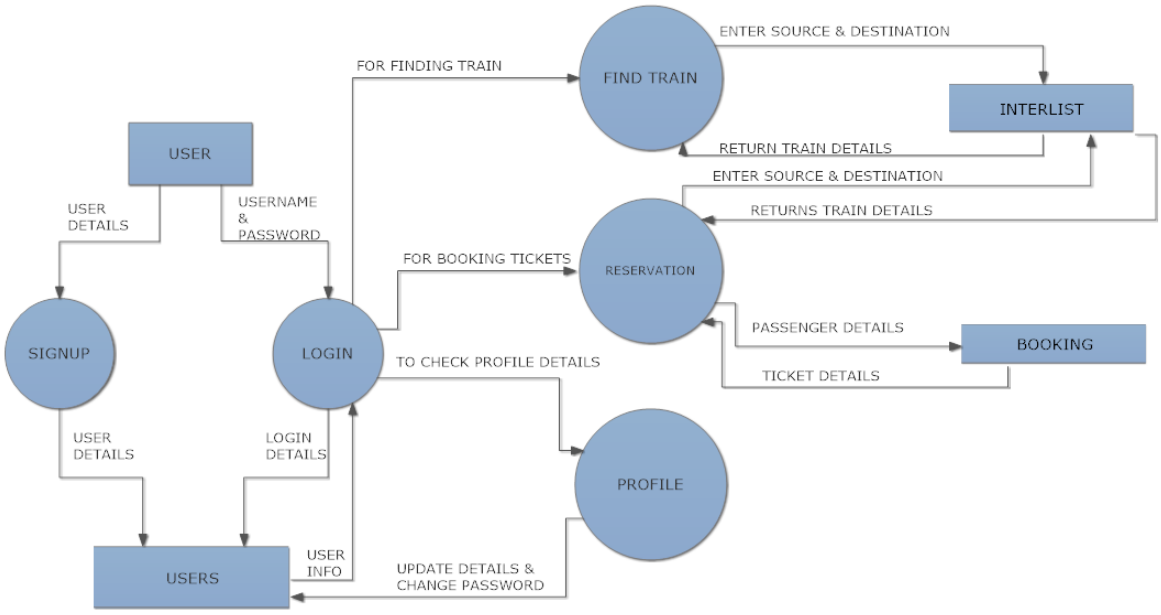


(4) Data store: Any store data but with no reference to the physical method of storing.

Level 0:

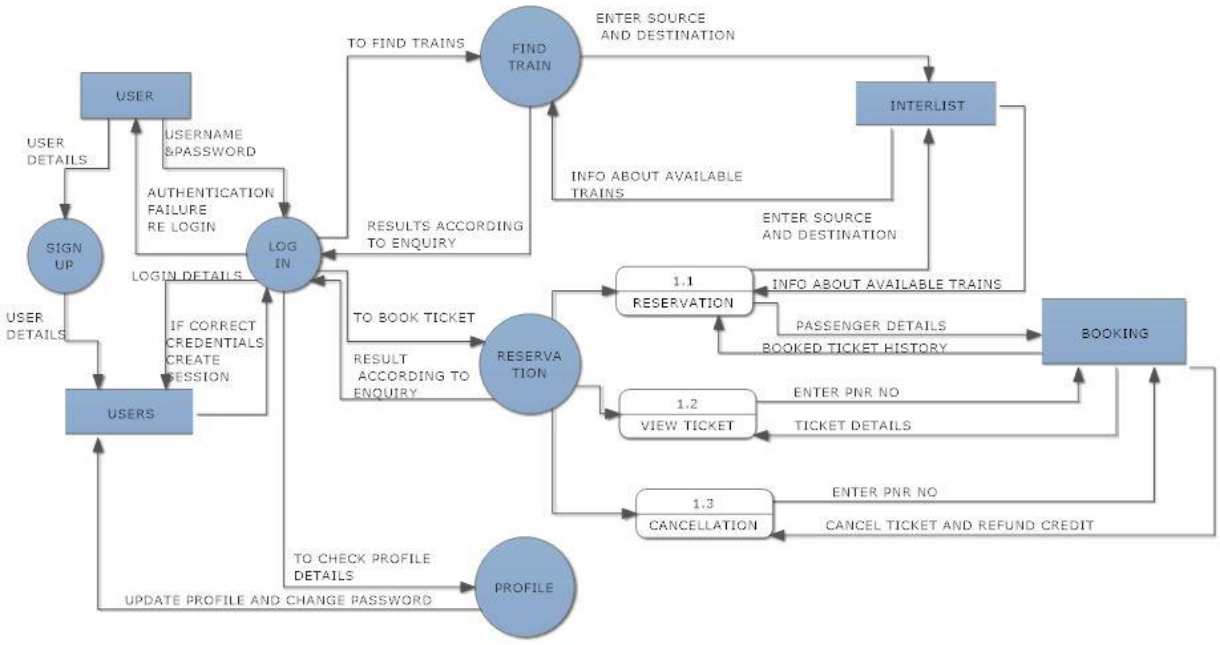
LEVEL 0 DATA FLOW DIAGRAM

Level 1:



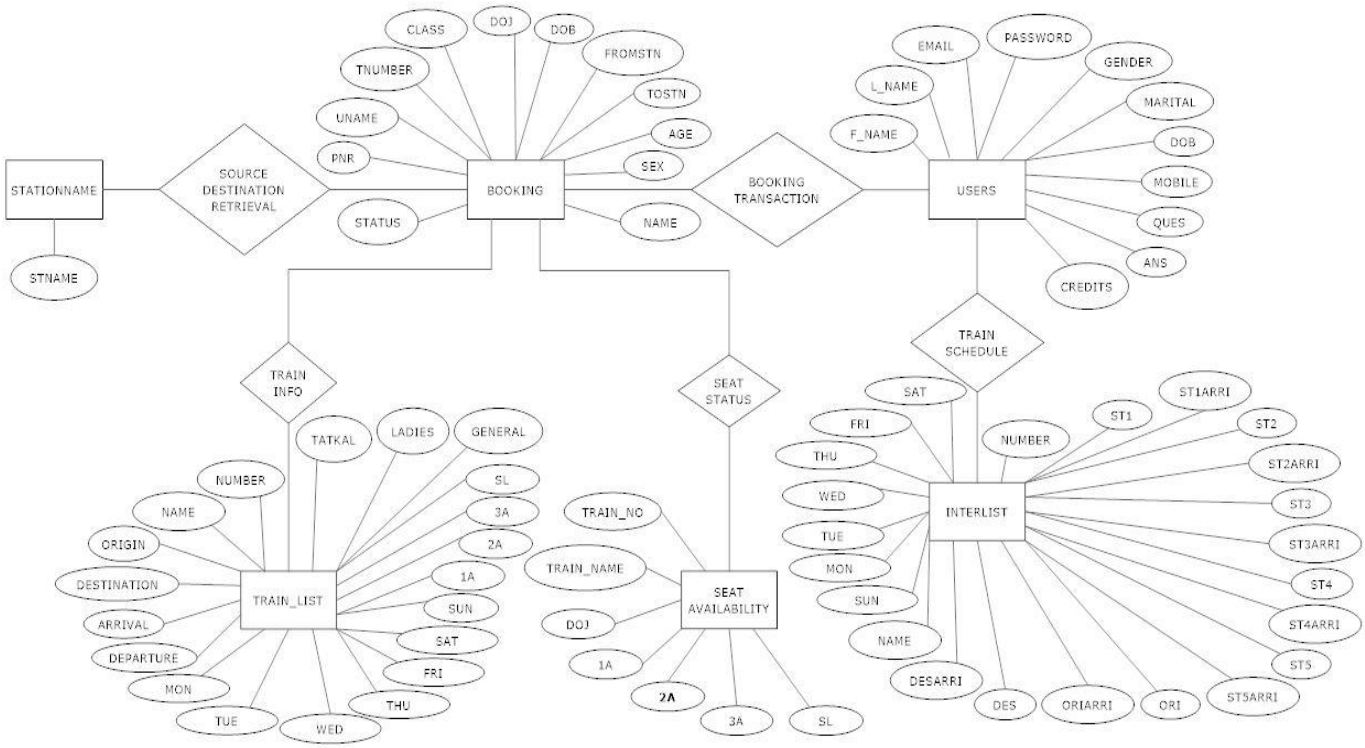
LEVEL 1 DATA FLOW DIAGRAM

Level 2:

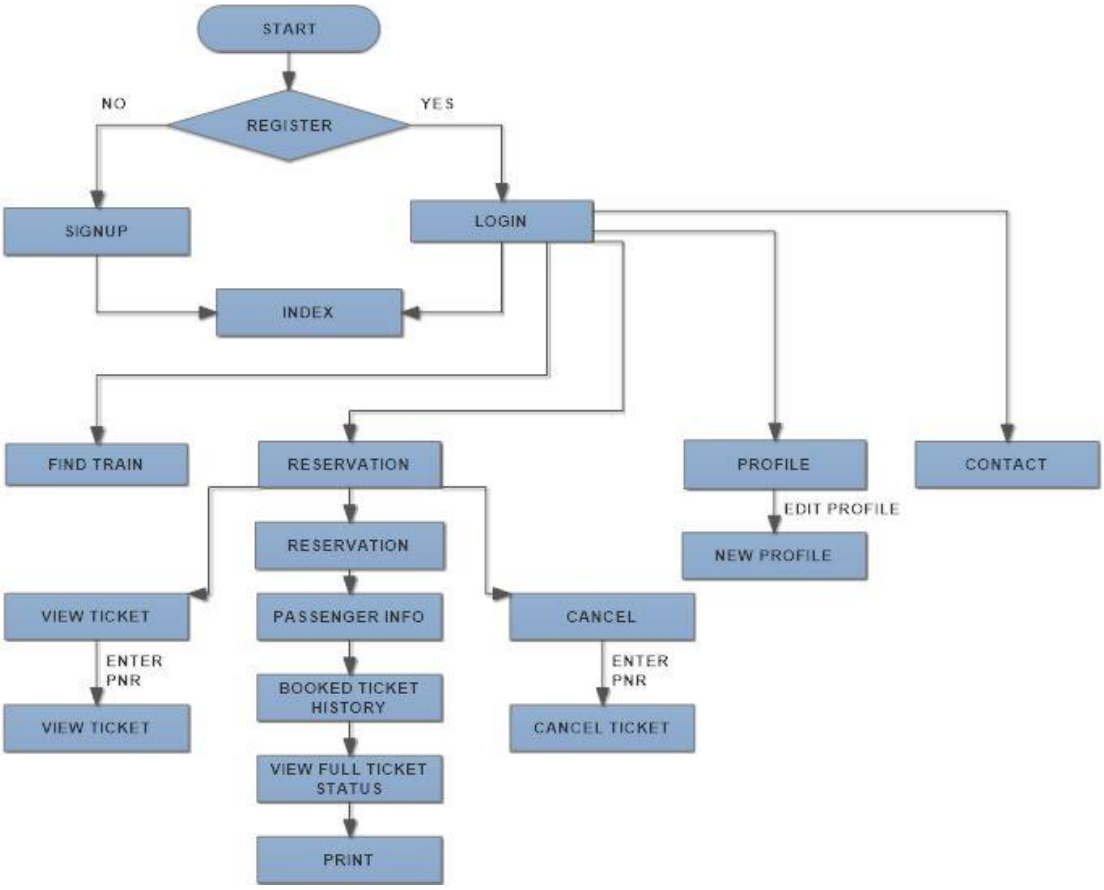


LEVEL 2 DATA FLOW DIAGRAM

ER DIAGRAM:


CLASS
DOJ
DOB
FROMSTN
TNUMBER
TOSTN
UNAME
AGE
PNR
SEX
STATUS
NAME
BOOKING
STATIONNAME
SOURCE DESTINATION RETRIEVAL
STNAME
BOOKING TRANSACTION
EMAIL
PASSWORD
GENDER
L_NAME
MARITAL
F_NAME
DOB
MOBILE
QUES
ANS
CREDITS
USERS
TRAIN INFO
TATKAL
LADIES
GENERAL
NUMBER
SL
NAME
3A
ORIGIN
2A
DESTINATION
1A
TRAIN_LIST
SUN
ARRIVAL
SAT
DEPARTURE
FRI
MON
THU
TUE
WED
SEAT STATUS
TRAIN_NO
TRAIN_NAME
SEAT AVAILABILITY
DOJ
1A
2A
3A
SL
TRAIN SCHEDULE
SAT
ST1ARRI
FRI
ST1
NUMBER
ST2
THU
ST2ARRI
WED
INTERLIST
ST3
TUE
ST3ARRI
MON
ST4
SUN
ST4ARRI
NAME
ST5
DESARRI
DES
ORIARRI
ORI
ST5ARRI

FLOW CHART:


START
REGISTER
NO
YES
SIGNUP
LOGIN
INDEX
FIND TRAIN
RESERVATION
PROFILE
CONTACT
EDIT PROFILE
NEW PROFILE
RESERVATION
VIEW TICKET
PASSENGER INFO
CANCEL
ENTER PNR
ENTER PNR
BOOKED TICKET HISTORY
VIEW TICKET
CANCEL TICKET
VIEW FULL TICKET STATUS
PRINT

9. PROJECT SCHEDULING:

SNo	Activity	Status	Responsibility	Week	Duration in Date
1	Information gathering	0%-7%	Team Members(2)	1st	02/02/18 – 04/02/18
2	Planning and scheduling	8%-15%	Do	1st & 2nd	05/02/18 – 09/02/18
3	Analysis & specification	16%-25%	Do	2nd & 3rd	10/02/18-17/02/18
4	Design	26%-40%	Do	3rd & 4th	18/02/18 – 26/02/18
5	Coding	40%-75%	Do	4th to 9th	26/02/18 – 30/03/18
6	Testing & validation	76%-82%	Do	9th to 10th	31/03/18 – 03/04/18
7	Implementation	83%-88%	Do	10th	04/04/18 - 09/04/18
8	Security checks	88%-94%	Do	10th to 11th	09/04/18 - 15/04/18
9	Extended feature adds	94%-100%	Do	11th to 12th	15/04/18-25/04/18
10	Submission	100%	Do	12th	30/04/18-30/04/18

9.1 Tabular representation of Project scheduling:

ACTIVITY		PERSON CONCERNED	WEEK	COMMENCE DATE
Information gathering		Team Members (2)	1st	02/02/18 – 04/02/18
Planning & scheduling	Pert chart		1st & 2nd	05/02/18 – 09/02/18
	Gnatt chart			
Analysis and specification			2nd & 3rd	10/02/18- 17/02/18
Design	Data Flow Diagram		3rd & 4th	18/02/18 – 26/02/18
	Flow Chart			
	Entity Relationship Diagram			
Coding	Website user interface coding		4th to 9th	26/02/18 – 30/03/18
	Creation of database and data insertion			
	Linking web interface to database			
Testing & validation			9th to 10th	31/03/18 – 03/04/18
Implementation			10th	04/04/18 - 09/04/18
Security checks for wrong input			10th to 11th	15/04/18- 25/04/18
Extended addition of features			11th to 12th	15/04/18- 25/04/18
Submission			12th	30/04/18- 30/04/18

9.2 GRAPHICAL REPRESENTATION OF THE ACTIVITY SCHEDULE

Gnatt Chart:

	Task	Assigned To	Start	End	Dur	%	2018 Feb	Mar	Apr	May
	⊖	TEAM MEMBERS(2)	2/2/18	30/4/18	61					
1	INFORMATION GATHERING	DO	2/2/18	6/2/18	3					
2	PLANNING & SCHEDULING	DO	5/2/18	9/2/18	5					
3	ANALYSIS & SPECIFICATION	DO	10/2/18	17/2/18	5					
4	DESIGN	DO	18/2/18	26/2/18	5					
5	CODING	DO	26/2/18	30/3/18	25					
6	TESTING & VALIDATION	DO	31/3/18	3/4/18	2					
7	IMPLEMENTATION	DO	4/4/18	9/4/18	4					
8	SECURITY CHECK	DO	9/4/18	15/4/18	5					
9	EXTENDED FEATURES ADDS	DO	15/4/18	25/4/18	8					
10	SUBMISSION	DO	30/4/18	30/4/18	1					

10. TESTING & VALIDATION

TESTING:

Test Criteria:

Software validation is achieved through a series of Black-box test that Demonstrate conformity with requirements. A test plan outlines the classes of test to be conducted and test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both the plan are satisfied, all behavioral characterizes are achieved performance.

Requirement are satisfied, all behavioral characteristics are achieved, all performance requirements are attained, documentation is correct and human engineering and other requirement are met.

After each validation test cases has been conducted. One of two possible condition exits:

1. The function or performance characteristics confirm to so edification and are accepting.

2. A deviation from specification is uncovered and a deficiency list is created.

Alpha and Beta Testing:

It is virtually possible for a software developer to foresee how the customer will really use the program. Instruction for use may be misinterpreted, strange combination of data may be regularly used, and output that seemed clear to the tester may be not intangible to a user in the field.

When customer software is built for one customer, a series of acceptance test are conducted to enable that customer to validate all requirements. Conducted by the end user rather than then software engineers, an acceptance. Test can range from an informal “test drive” to a planned and systematically executed series to test. Infect acceptance testing can be conducted over a period of week or months, thereby uncovering cumulative errors that might degrades the system. Most software product builders use a process called alpha and beta testing to uncover errors that only the end user seems able to find.

The beta testing is conducted at one or more customer sites by the end user of the customer. Unlike alpha testing, the developer is generally not present. Therefore the beta test is a “live” application of the software in a environment that cannot be controlled by the developer.

Testing Techniques and Testing Strategies

Software Testing Techniques

The importance of software testing to software quality cannot be overemphasized. Once source code has been generated, software must be tested to allow errors to be identified and removed before delivery to the customer. While it is not possible to remove every error in a large software package, the software engineer’s goal is to remove as many as possible early in the software development cycle. It is important to remember that testing can only find errors; it cannot prove that a program is bug free. Two basic test techniques involve testing module input/output (black box) and exercising internal logic of software components (white-box). Formal technical reviews cannot find allow software defects, test data must also be used. For large software projects, separate test teams may be used to develop and execute the set of test cases used in testing.

Software Testing Objectives

- Testing is the process of executing a program with the intent of finding errors.
- A good test case is one with a high probability of finding an as-yet undiscovered error.
- A successful test is one that discovers an as-yet-undiscovered error.

Software Testing Principles

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The Pareto principle (80% of all errors will likely be found in 20% of the code) applies to software testing.
- Testing should begin in the small and progress to the large.
- Exhaustive testing is not possible.
- To be most effective, testing should be conducted by an independent third party.

Software Testability Checklist

- Operability (the better it works the more efficiently it can be tested)
- Observably (what you see is what you test)
- Controllability (the better software can be controlled the more testing can be automated and optimized)
- Decomposability (by controlling the scope of testing, the more quickly problems can be isolated and retested intelligently)
- Simplicity (the less there is to test, the more quickly we can test)
- Stability (the fewer the changes, the fewer the disruptions to testing)
- Understandability (the more information known, the smarter the testing)

Good Test Attributes

- A good test has a high probability of finding an error.
- A good test is not redundant.
- A good test should be best of breed.
- A good test should not be too complex or too simple.

Test Case Design Strategies

White box testing uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs.In electrical hardware testing, every node in a circuit may be probed and measured; an example is in circuit testing (ICT).

Since the tests are based on the actual implementation, if the implementation changes, the tests probably will need to change, too. For example ICT needs updates if component values change, and Black-box or behavioral testing (knowing the specified function a product is to perform and demonstrating correct operation based solely on its specification without regard for its internal logic)

White-box or glass-box testing (knowing the internal workings of a product, tests are performed to check the workings of all independent logic paths)

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually

functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure.

This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance. The higher the level, and hence the bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested.

Functional testing devices like power supplies, amplifiers, and many other simple function electrical devices is common in the electronics industry. Automated functional testing of specified characteristics is used for production testing, and part of design validation needs modified/new fixture if the circuit changes. This adds financial resistance to the change process, thus buggy products may stay buggy. Automated optical inspection (AOI) offers similar component level correctness checking without the cost of ICT fixtures; however changes still require test updates. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also test paths between units during integration, and between subsystems during a system level test. Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed.

In computer programming, **unit testing** is a software verification and validation method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest

unit is a method, which may belong to a base/super class, abstract class or derived/child class.

Unit testing can be done by something as simple as stepping through code in a debugger; modern applications include the use of a test framework such as xUnit. Ideally, each test case is independent from the others: substitutes like method stubs, mock objects, fakes and test harnesses can be used to

assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its requirements and behaves as intended. Its implementation can vary from being very manual (pencil & paper) to being formalized as a part of build automation.

Integration testing (sometimes called Integration and Testing, abbreviated I&T) is the activity of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items",i.e. assemblages (or groups of units), are exercised through their interfaces using Black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process

Different levels of Test

Testing occurs at every stage of system construction. The larger a piece of code is, when defects are detected, the harder and more expensive it is to find and correct the defects. The different levels of testing reflect that testing, in the general sense, is not a single phase of the software lifecycle. It is a set of activities performed throughout the entire software lifecycle. In considering testing, the activities described are shown in the figure. The activities after implementation are normally the only ones associated with testing.

SNAPSHOT:

Welcome screen

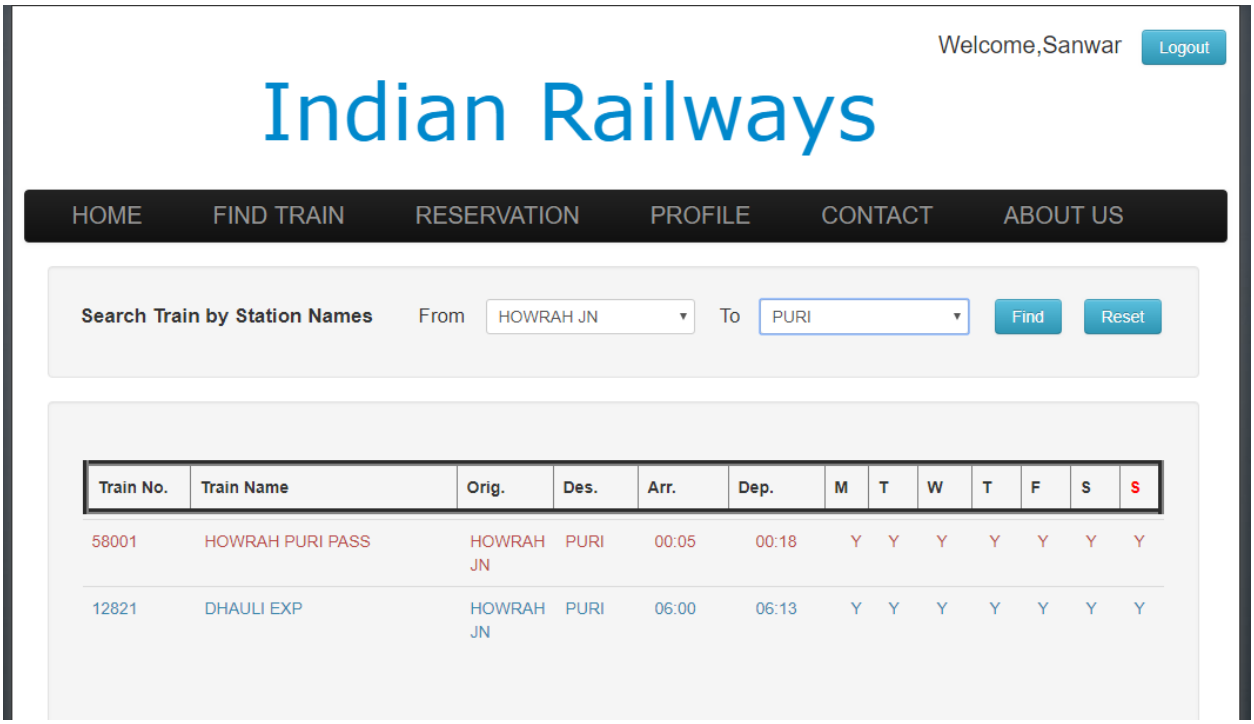
Signup


Indian Railways
HOME
FIND TRAIN
RESERVATION
PROFILE
CONTACT
ABOUT US
Signup
First Name *
Enter the First name
Last Name *
Enter the Last name
Email ID *
Enter the valid email id
Password *
Enter the password
Confirm Password *
Re-type the password
Gender *
MALE
Marital Status *
Married
Date of Birth *
08-May-2000
Mobile No. *
+91
Security Question *
What is your pets name ?
Your Answer *
Enter the your answer
Enter Credits *
Add atleast 5000 credits
submit
Reset

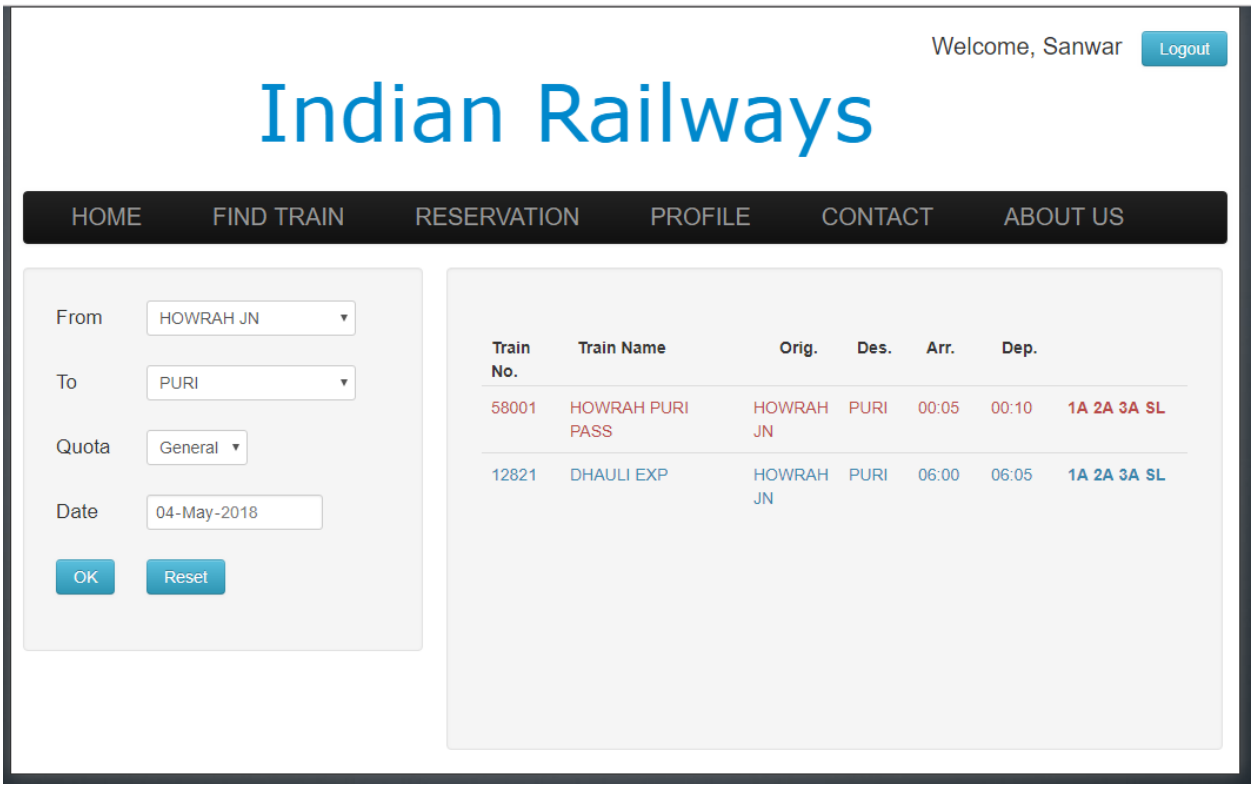
Login


Indian Railways
HOME
FIND TRAIN
RESERVATION
PROFILE
CONTACT
ABOUT US
Username
Username
Password
password
Login
Haven't signed up? Signup below
Signup

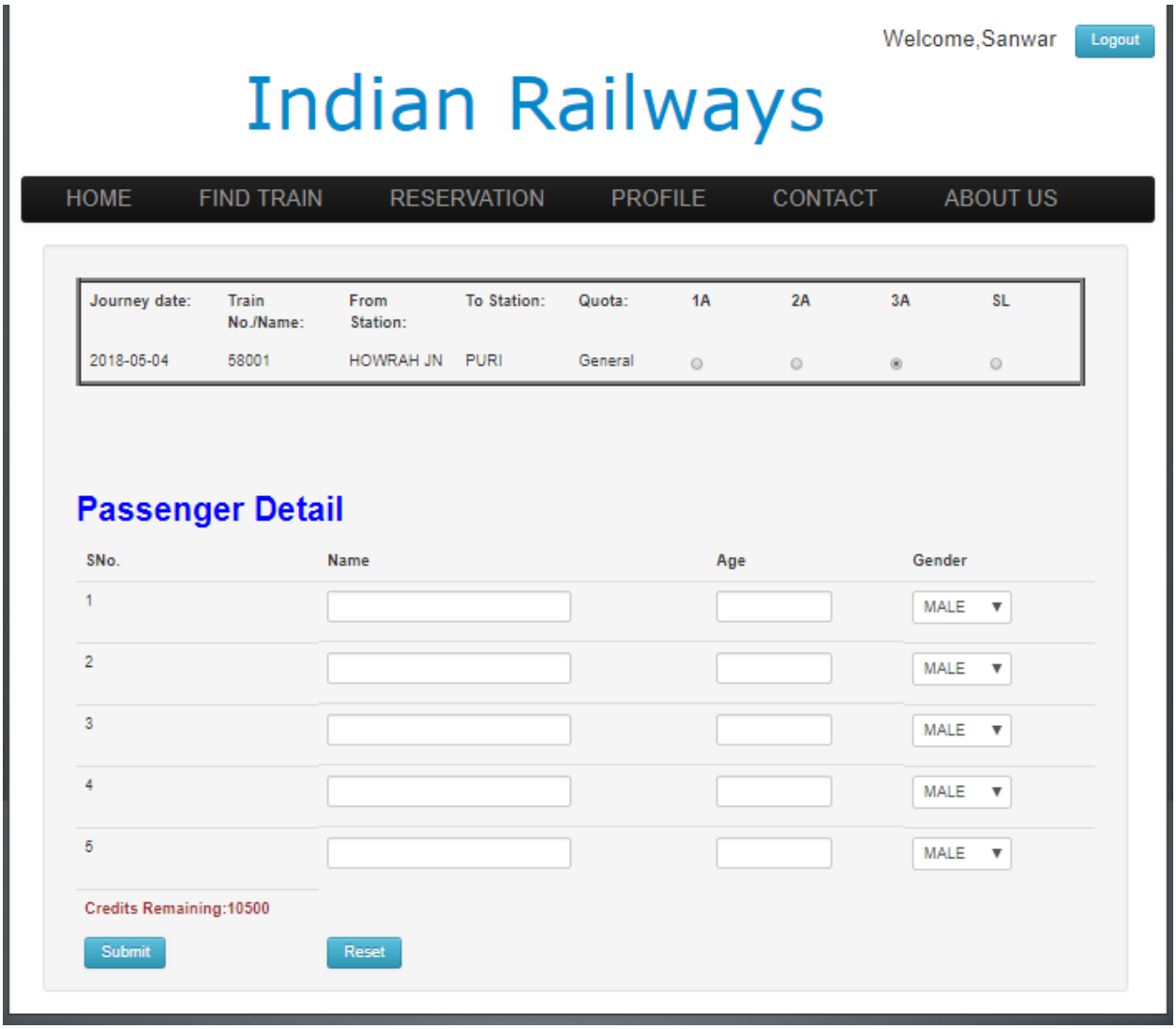
Train searching



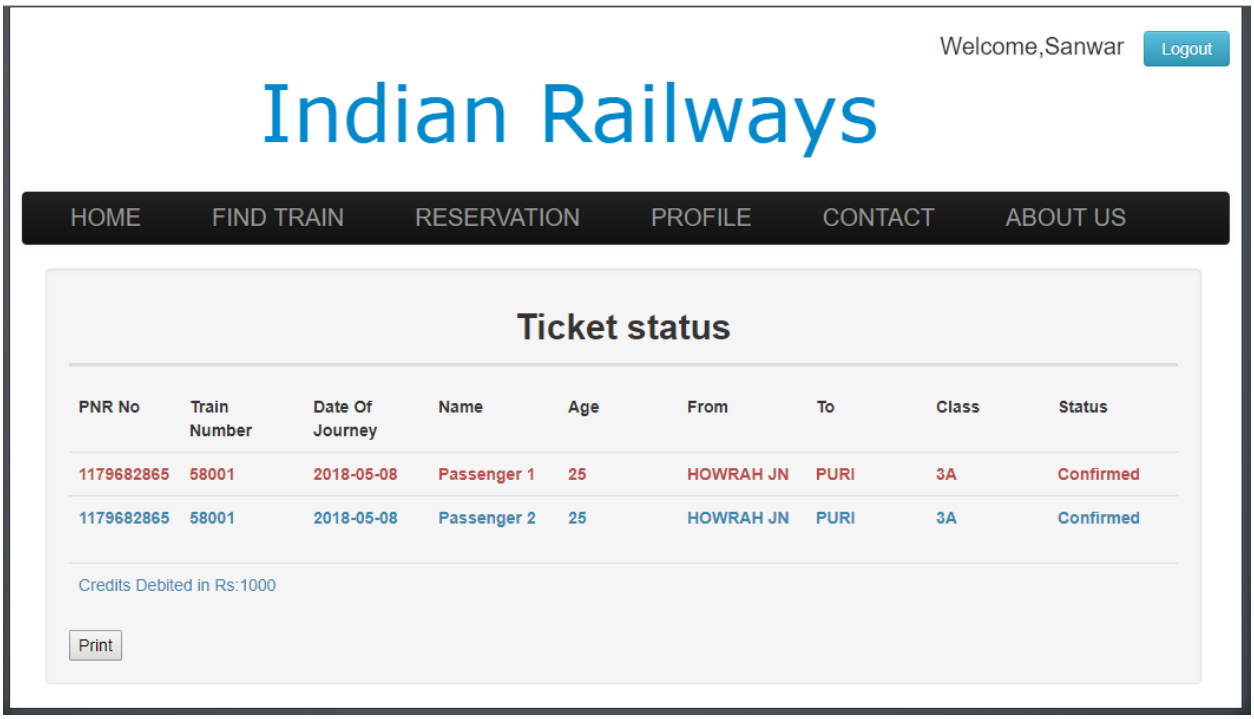
Reservation



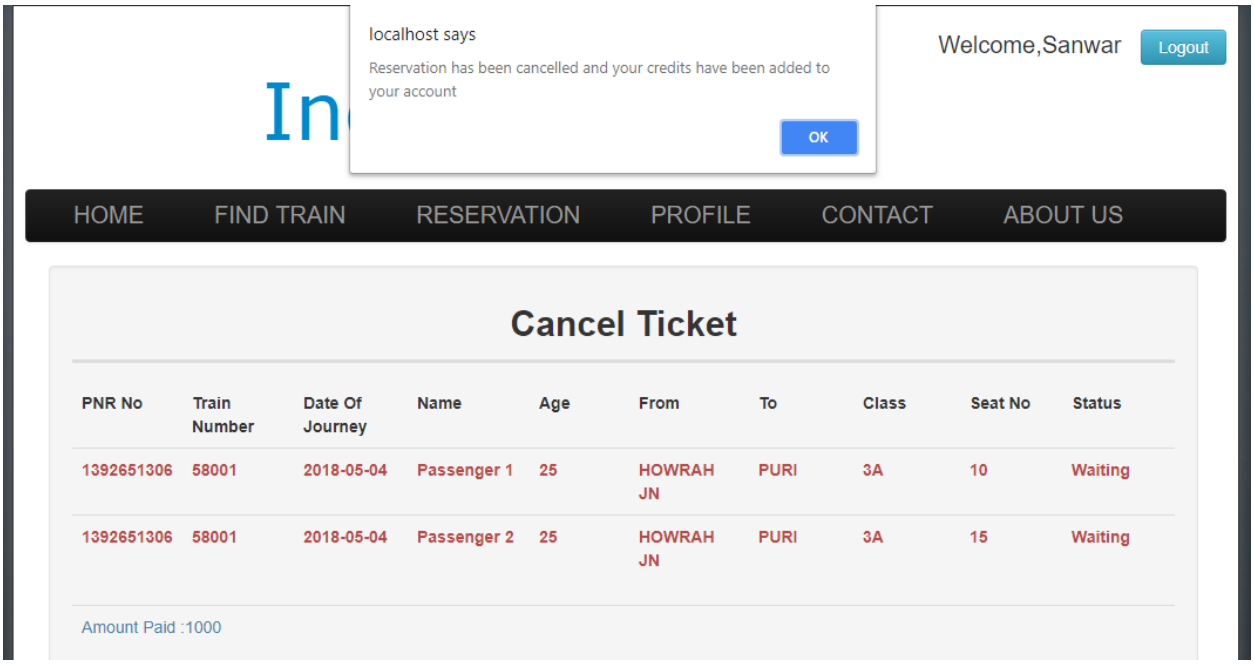
Passenger Information



View Ticket



Cancellation



FUTURE SCOPE:

The software can reduce significant amount of work load on the staff as well as make the reservation procedure hassle free to the customers, along with it can help saving time and money. The customers don't need to visit any railway station for the tickets issues, these matters can be solved in their homes itself.

The railways can then look into other perspective of improving the efforts to make the railways clean and safe for passengers and the online payments can refund the customers accurately to decimals.

This system is essential for a country like India where thousands of people travel every day and this creates a chaos in the reservation system, with proper maintenance, and addition of few other features as required in software this chaos can be significantly contained to manageable scale. Therefore implementation of this software is a necessity.

CONCLUSION:

The project online railway reservation system had a very essential value in terms of travelling, it makes the hassle of running to a train station for booking tickets to right in the house. The projects enables the users to check time table for the trains available, book tickets, check the ticket status after booking using the PNR no generated. The project has a lot of potential and can even enable paperless travelling in future. The project requires huge resource investment to cater the demand properly, so there would be issues once in a while, for that situation not to arise proper maintenance and innovation is required from time to time. As the number of travelers are not easy to count on hands these days, there's an essential requirement of these kind of software, they benefit the traveler, they benefit the staff and they avoid chaos at the counter and makes travelling a lot easier.

BIBLIOGRAPHY:

- https://indianrail.gov.in
- https://w3schools.com
- https://php.net/manual/en/index.php