# 1.1 ABOUT THE PROJECT

The project named Online Turf Booking System is to create a fully computerized system implemented to book online, store the details of various timeslots and also to reduce the manual process of managing the turf. The customers can book the turf by registering in the system by giving the requested credentials. After the customer registration is complete, the customer can view the details of all the timeslots that are available for booking on the respective date selected by the user. The system offers the facility to book turf for six days in advance from the present date. If the customer wishes to book a slot, they can book that particular timeslot by paying the amount either through debit or credit card by entering the card details. Once the payment is completed the customer is provided a QR code which is to be shown at the centre to avail their reservation. The main aim of the project is to provide an effective working platform to computerize the whole system into electronic form in order to decrease the manual work.

The proposed project on Online Turf Booking System is an effort to solve the various problems in booking a turf. The implementation of the project establishes a systematic and reliable service. Using this system, one can make reservations through internet.

The project is computerizing an Online Turf Booking System that has a vast collection of timeslots and deals with its various transactions. The customer can avoid delay by using the system. It is very smooth, efficient and fast system. The current states of the system can be understood by using the report form which is very helpful for the administrator. The manual system had many limitations, however through the process of computerization, the system overcomes those limitations.

**Advantages for the Project**

1. It allows the customers to create their own account and stores their details in the database.
2. Once the customer books the turf successfully, a QR code will be shown confirming their reservation.
3. It makes the easy for the customers to know the availability of the turf on a respective date, their price.
4. It makes it easy for the customers to know the availability of the turf and its price.

# 2.1 SYSTEM STUDY

## 2.1.1 Existing system

In existing system, to manage the soccer turf we use a manual system. In the present system all the activities are performed manually. All data entry is performed by writing data into the book, paper documents. The record is prepared manually, so there is a chance for occurring errors and the calculations are not so accurate. When there is need for retrieving details searching is unavoidable, this is a difficult task searching the records manually.

The existing system uses manual method for whole process. This requires a lot of hard work and time consumption to complete the task. This may include human errors. In this existing system, it is difficult to retrieve some particular information. Also, all the records are stored manually and it is tedious task. As a result, the security of these records is always a challenging task. Hence, the computerization of the system of record maintenance is the only solution to reduce the shortcomings of existing system.

# Disadvantages of existing system

* Time consuming and improper planning and scheduling.
* High storage space.
* Less interactive.
* Requires more man power.
* High maintenance costs.

## 2.1.2 Proposed System

The proposed system is interactive, highly user friendly and designed exclusively for the Soccer Turf. The Online Turf Booking System is a database system used to store the information regarding customer details, price details, booking details etc.

All the operations and activities related to The Online Turf Booking System can be carried out efficiently. The system manages a well-organized database for storing the resources that they are providing by theSoccer Turf. This helps us to eliminate the entering of invalid data. Most problems of manual system can be solved by this system.

The computerization of the system allows the easy maintenance of the details. Large amount of data can be stored easily. In addition, updating and other changes can be done easily. The information can be retrieved with high speed and accuracy. The use of GUI oriented software makes the system user friendly.

# Advantages of Proposed System

* High processing speed.
* Easy to retrieve old records by using search feature
* We can analyze customer details, booking details, price details.
* The system shall provide the capability to back up the database.
* Minimal errors.
* Greater portability.
* User friendly.
* Reduced workload.

## 2.1.3 Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, time and effort spent on it.

Study lets the developers to see the future of the project and its usefulness. Finding out whether a new system is required or not.

The study is carried out to the best system that meet performance requirement.

This entails identification, description and evaluation of candidate system and selection of the best system for the job. It simply identifies whether the proposed system is feasible to the organization or not.

There are three aspects in the feasibility study portion of the preliminary investigation

i) Technical feasibility ii) Economic feasibility iii) Operational feasibility

### 2.1.3.1 Technical Feasibility

The Online Turf Booking System must be evaluated from technical viewpoint first. The assessment of this feasibility must be based on an outline design of the system requirements in the terms of input, output, programs and procedure having identified an outline system, the investigation must go on to suggest the type of equipment, required method of developing the system, method of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed with latest technology. There are only minimal constraints involved in this project

### 2.1.3.2 Economic Feasibility

Here an evaluation of development cost is weighted against the ultimate income or benefit derived from the developed system. The cost for the development of the project has been evaluated and we want to check that the cost does not exceed beneficial cost of the system. The economic and financial analysis is used for evaluating the effectiveness of the candidate system. This project also undergone economic feasibility study and found that it is feasible. So, the cost for development does not exceed its beneficial cost. This brought to as the conclusion that the system is economically feasible in the context.

### 2.1.3.3 Operational Feasibility

In operational feasibility the entire application is checked whether the system will be used if it is developed and implemented. Also, it is checked whether there will be resistance from user that may undermine the possible application benefits. There is no barrier for implementing the system. The system also helps to access the information immediately as need arises. Thus, the system is found to be operational feasible.

# 2.2 USER CHARACTERISTICS

This software has two users

1. Administrator
2. Customer

**Administrator:**

Administrator is the primary user who has the most or maximum control over the software. Administrator administrates over the entire activities of the system and has full control over what all happens in the Shop. He/she is the only user who can block slots, update pricing and make changes to the system.

**Customer:**

The customer will have little privileges when compared to the administrator. They can book the turf and reserve them. They can also view their previous booked turf details.

# 2.3 SYSTEM SPECIFICATION

## 2.3.1 Hardware Specification

The selection of hardware and software configuration is very important task related to system development.

|  |  |
| --- | --- |
| Processor | Intel Pentium IV (3.0 GHz) or above |
| RAM | 1 GB |
| Hard Disk | 80 GB and above |
| Key Board | Normal or multimedia |
| Monitor | 15’’CRT or LCD monitor |
| Mouse | Compatible Mouse |

## 2.3.2 Software Specification

|  |  |
| --- | --- |
| Operating System | Windows |
| Front | PHP |
| Back End | SQL Server 2008 |

## 2.3.3 About The Software Tools

FRONT END SPECIFICATION: PHP

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.PHP is an embedded scripting language that is excellent for creating dynamic Web sites based on database content or different characteristics of browsers. It is available when you have a [Departmental (Web Central) Publishing account,](http://www.utexas.edu/learn/puboptions/dept.html) a [Faculty Publishing account,](http://www.utexas.edu/learn/puboptions/faculty.html) a [Student Organization Publishing account,](http://www.utexas.edu/learn/puboptions/org.html) or if you use [Custom Web Publishing.](http://www.utexas.edu/its/customweb)

# Features

* Allows you to build templates to ease site maintenance
* Serve different content to users based on their browser, IP address, date and time, or numerous other characteristics
* Enables connections with databases such as [MySQL](http://www.utexas.edu/its/mysql)
* Build discussion forums or Web-based email programs

# Characteristics of PHP

* Objected Oriented
* Complied and Interpreted
* Portable
* Distributed Secure BACK END SPECIFICATION: SQL Server 2008

SQL Server 2008 is an integrated database management system and analysis solution that delivers increased security, scalability and availability to enterprise data and analytical applications, while making them easier to build, deploy and manage. It is comprehensive software that enables to reliably manage mission – critical information and confidently run today’s increasingly complex business applications. SQL Server 2008 allows gaining greater insight and achieving faster results for a competitive advantage. The key capabilities of SQL Server 2008 are the following:

High Availability: Ensures business continuity with the highest levels of system availability through technologies that protect data against costly human errors and minimize disaster recovery downtime.

Performance and Scalability: Deliver an infrastructure that has proven record in handling today’s large amounts of data and critical enterprise workloads.

Security: Provides a secure environment to address privacy and compliance requirements with built in features that protect data against unauthorized access.

Manageability: Manages infrastructure with automated diagnostics, tuning and configuration to reduce operational costs while reducing maintainance and easily managing very large amounts of data.

Developer Productivity: Build and Deploy critical business ready applications more quickly by improving developer productivity and reducing project lifestyle times.

Business Intelligence: Gain deeper insight into the business with integrated comprehensive analysis and reporting for enhanced decision making.

# 3. SYSTEM MODELING

The most creative and challenging phase of the system development is system design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fullfil. The first step is to determine how the output is to be produced and in what format. Second input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing. The point is to choose such an environment in which we will be able to operate within a convenient and easy way. The most creative and challenging phase of the system development is system design. It provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The analyst should understand the requirements of the user and develop the system according. Design goes through the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The application program as an interface between the users and the database should be an accurate reflection of the database on the screen; hence a well analyzed and defined structure is needed. The user interface should be easy to understand and operate on for the users. The first step is to determine how the output is to be produced and in what format it has to be produced. Second, input data along with the master files have to be designed to meet the requirements of the proposed output.

The analyst must ensure that the interaction between the user and the interface is simple to understand. To ensure that everything works properly and as it has been expected, test performances have to be done upon the system functionality. Testing plays an important role in identifying any minor errors after system design and it will be corrected.

## 3.1 MODULES AND DESCRIPTION

The Online Turf Booking System is a web-based booking system which provides us facilities to manage the activities taking place in a Soccer Turf. This system is developed to manage the work flow in a Soccer Turf. There are 4 modules in this project. They are:

1. Customer Registration
2. Timeslot Management
3. Booking Management
4. Payment Management

# 1. CUSTOMER REGISTRATION

This module deals with the registration of the customer. Customer registration module also deals with adding new customers to the system, updating the details of the existing customer, searching for a particular customer and maintaining the status of the customer being registered to the system. The personal details of the customer are also being added to the system.

# 2. TIMESLOT MANAGEMENT

This module deals with the managing of the timeslot selected by the customer .Here the customer can perform the following task such as selecting the date and time period and also view the price allocated for the respective schedule whereas only the admin can block slots, edit their price. Rescheduling or cancellation of a slot that is booked is not permitted.

# 3. BOOKING MANAGEMENT

This module deals with booking of the turf with respect to the selected timeslot and is preceded to payment to complete their booking.

# 4. PAYMENT MANAGEMENT

In this module the customer can make payment and once the payment is done the booking status is updated. The payment mode preferred is through credit or debit cards.

## 3.2 DATA FLOW DIAGRAM (DFD)

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are central tool and the basis from which the other components are developed. The transformation of data from input to output, trough processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

A DFD is also known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

In the DFD, there are four symbols

Process that transforms dataflow

Source or Destination of data

Data store

Data flow

## Rules for drawing data flow diagrams

Establish the context of the data flow diagram by identifying all of the net input and output data flows.

Rule 2: Select a starting point for drawing the DFD.

Rule 3: Give meaningful labels to all data flow lines.

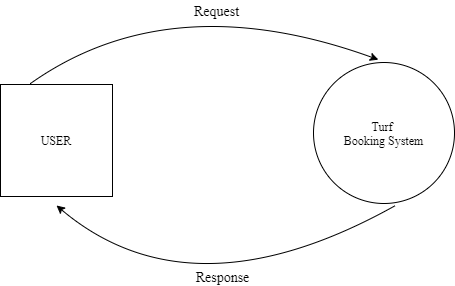
Rule 4: Label all processes with action verbs that relate input and output data flows.

Rule 5: Omit insignificant functions routinely handled in the programming process.

Rule 6: Do not include control or flow of control information.

Rule 7: Do not try to put too much information in one DFD. Rule 8: Be prepared to start over

## Level 0 DFD Showing Turf Booking System

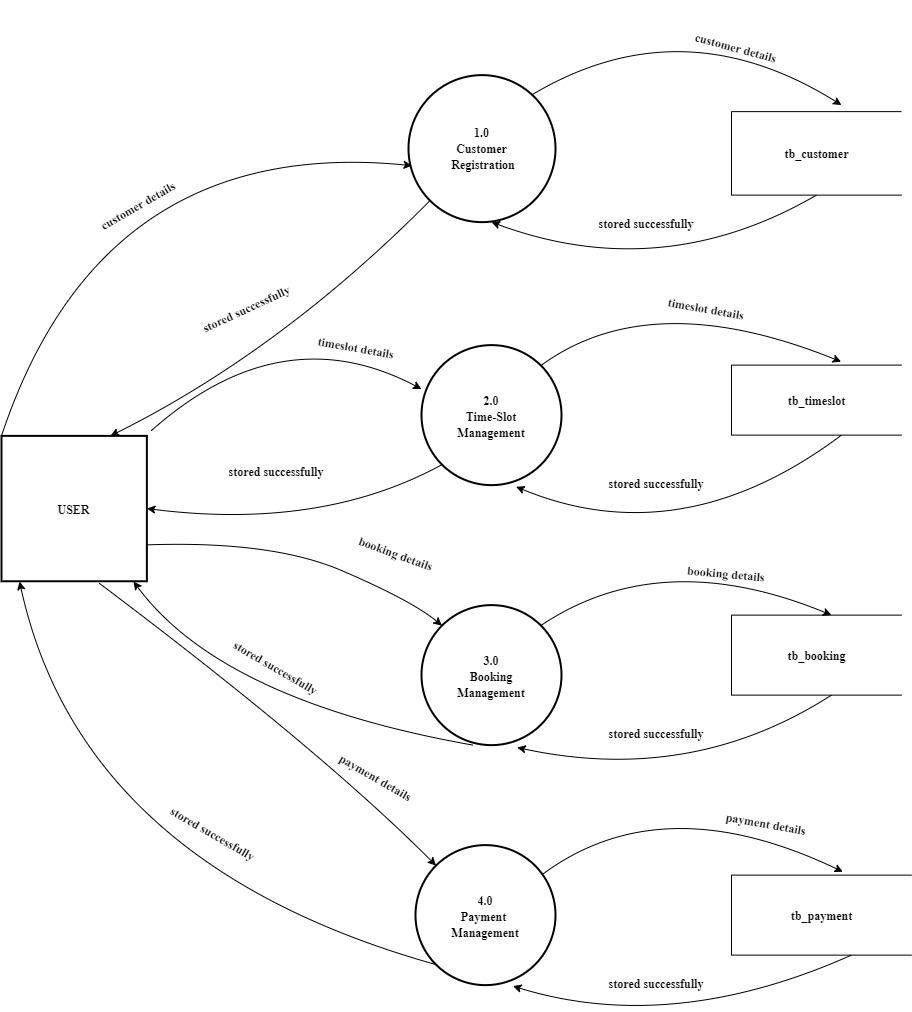


**Level 1 DFD**

**Showing**

**Turf Booking**

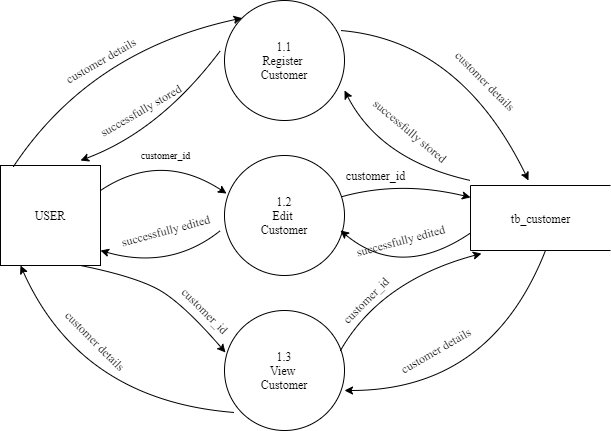
**System**



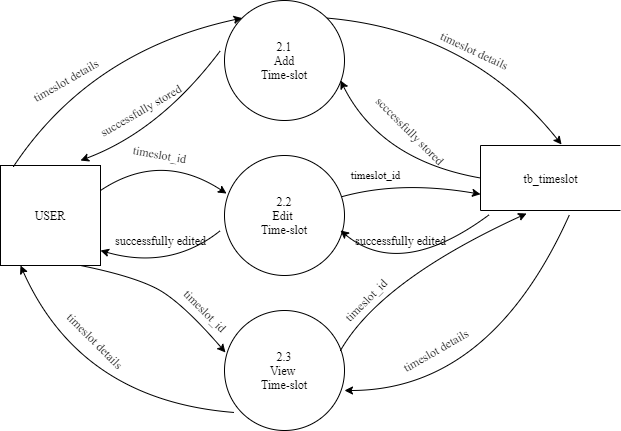
**Level 2 DFD**

**Showing**

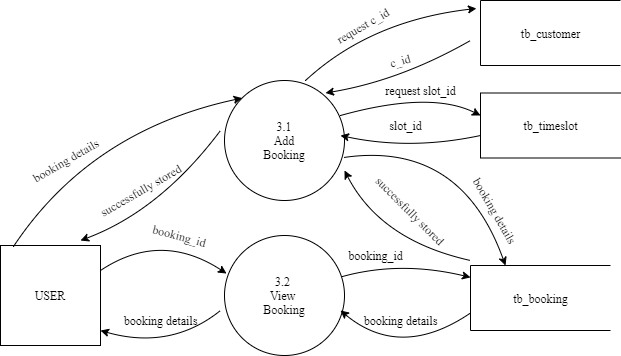
**Customer Registration**



**Timeslot**

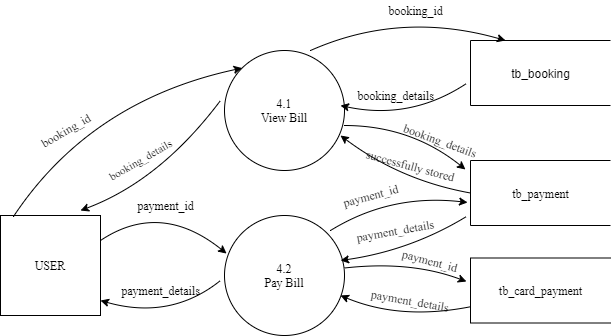


**Booking**



**P**

**ayment**



## 3.3 ENTITY RELATIONSHIP DIAGRAM

The ER model is a conceptual data model that views the real world as a construct of entities and associations or relationships between entities. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. The ER modelling technique is frequently used for the conceptual design of database applications and many database applications and many database design tools employ its concepts.

Entity Type

Weak Entity Type

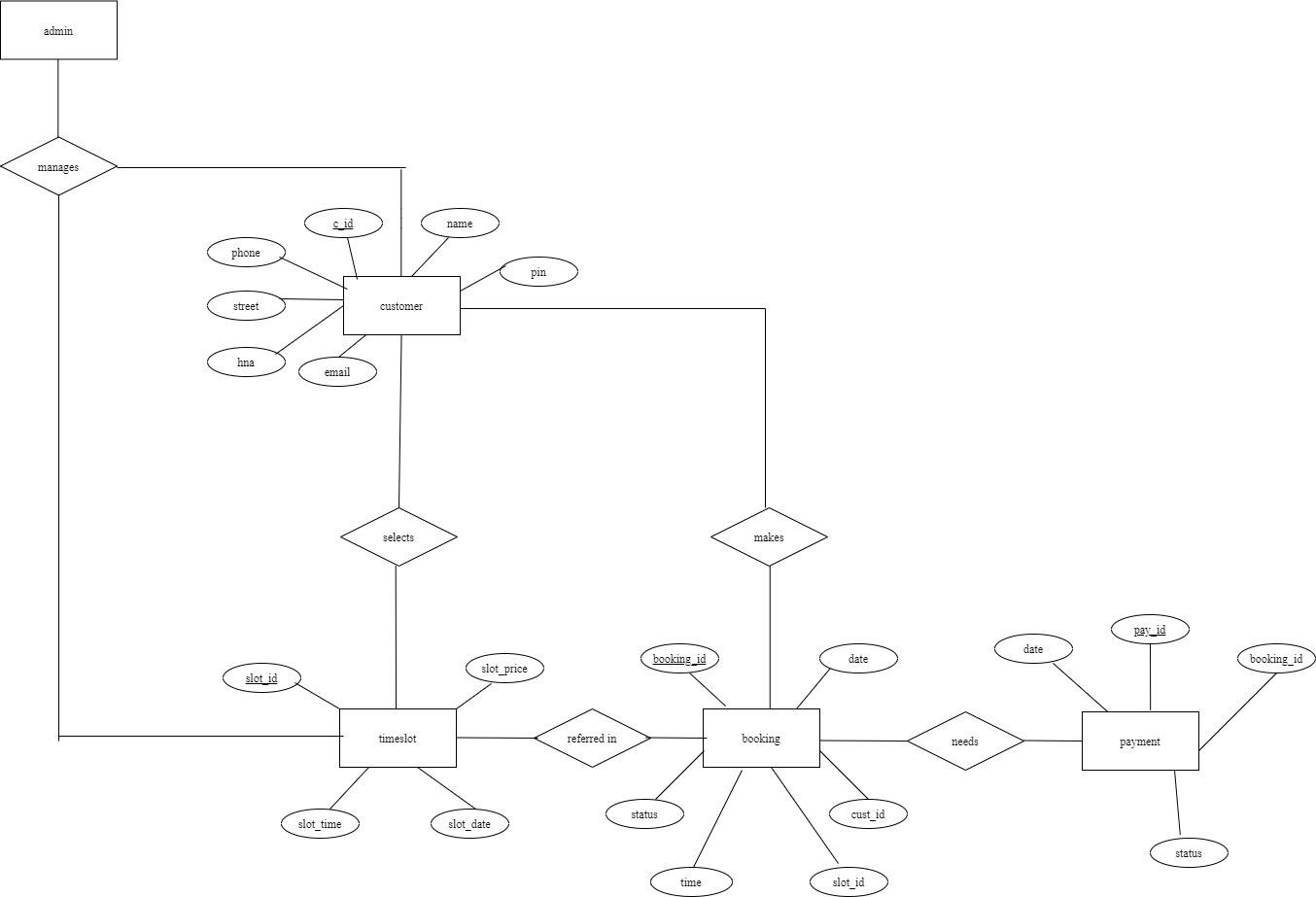
Relationship Type

Attribute

Key attribute

Multi-valued Attribute

**ER DIAGRAM**



## 4.1 INPUT DESIGN

Input design is the process of converting a user-oriented description of the inputs to a computer-based business system into a programmer-oriented specification. The quality of system input determines the quality of system output. Input specification describes the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of errors.

The input design also determines whether the user can interact efficiently with the system. Input design requires consideration of the needs of the data entry operator. Three data entry considerations are:

* The field length must be documented
* The sequence of fields must match the sequence of the fields on the source document.
* The data format must be identified to the data entry operator.

In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry can be controlled by input design. Input design is the process of converting user-oriented inputs to a computerbased format. There are three major approaches for entering data into the computer.

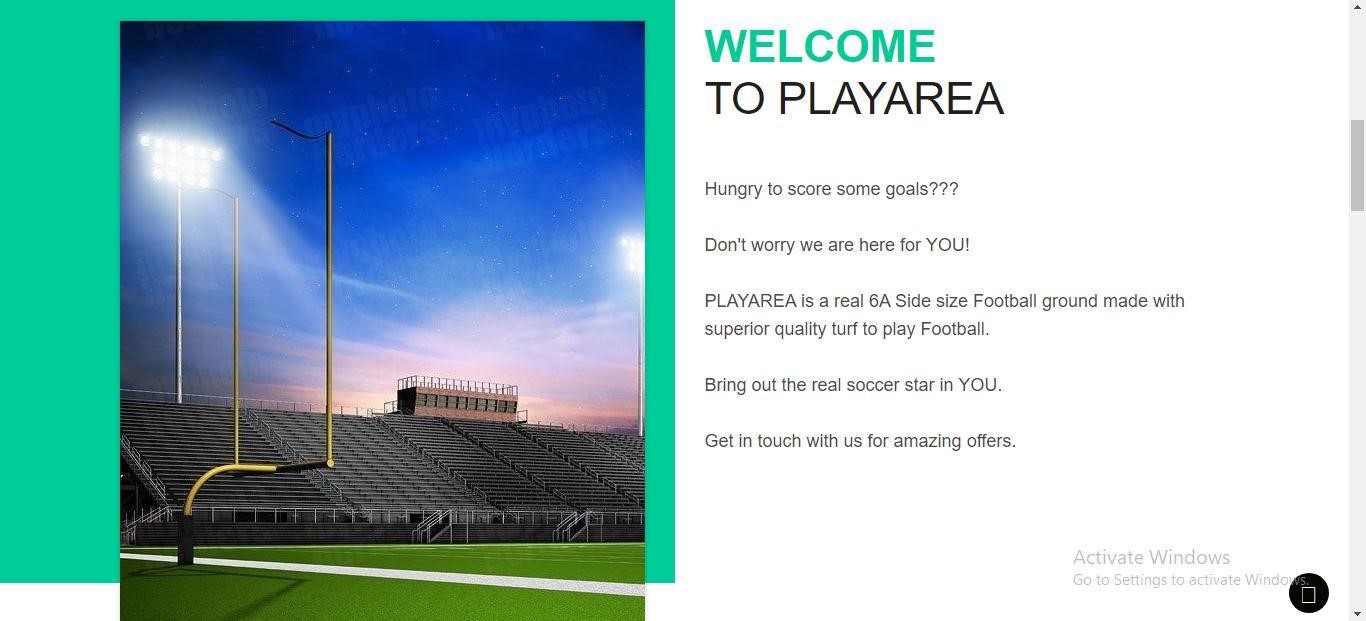
They are menus, formatted forms and prompts. A menu is a selection list that simplifies computer data access or entry. Instead of remembering what to enter, the user choices from a list of option.

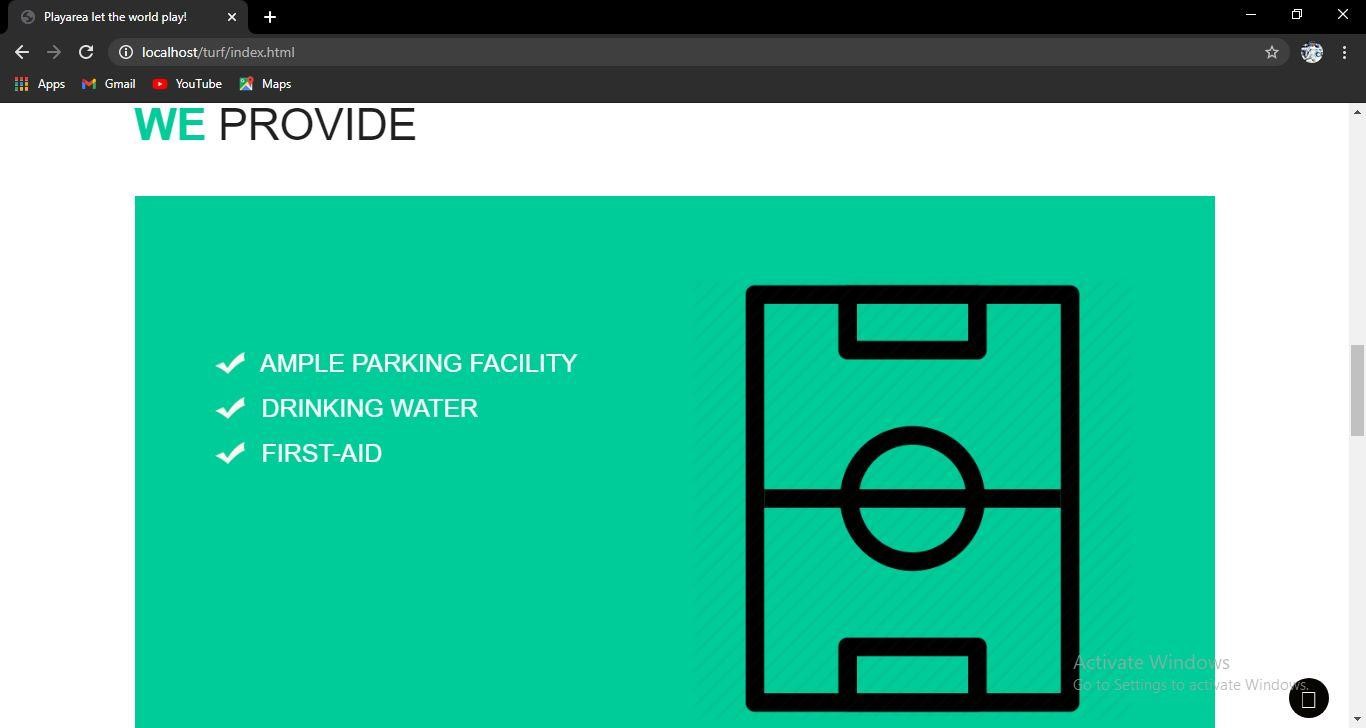
A formatted form is a pre printed form or a template that request the user to enter data in appropriate location. It is a fill-in-the-blank type form. The form is flashed on the screen as a unit. In prompt the system displays one enquiry at a time, asking the user for a response.

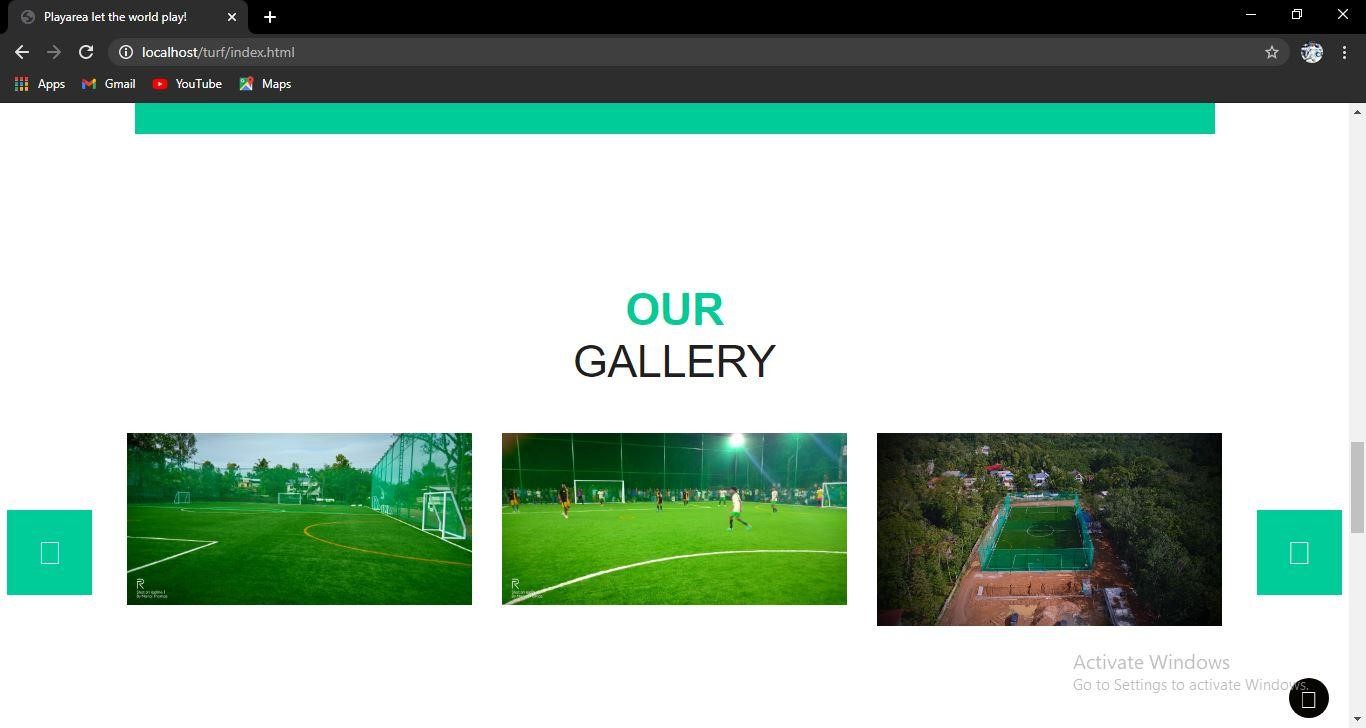
## Home Page

**Description:** This form shows the Home page for all the users.



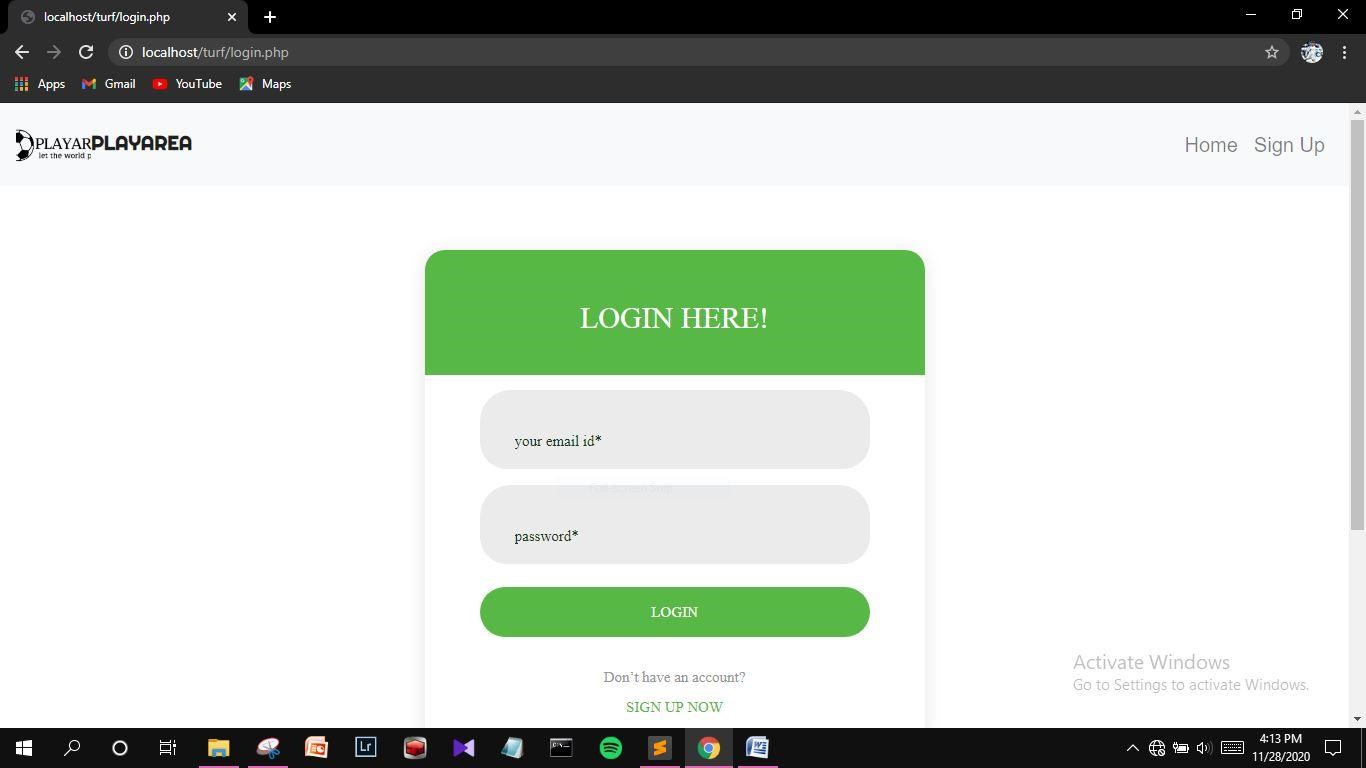






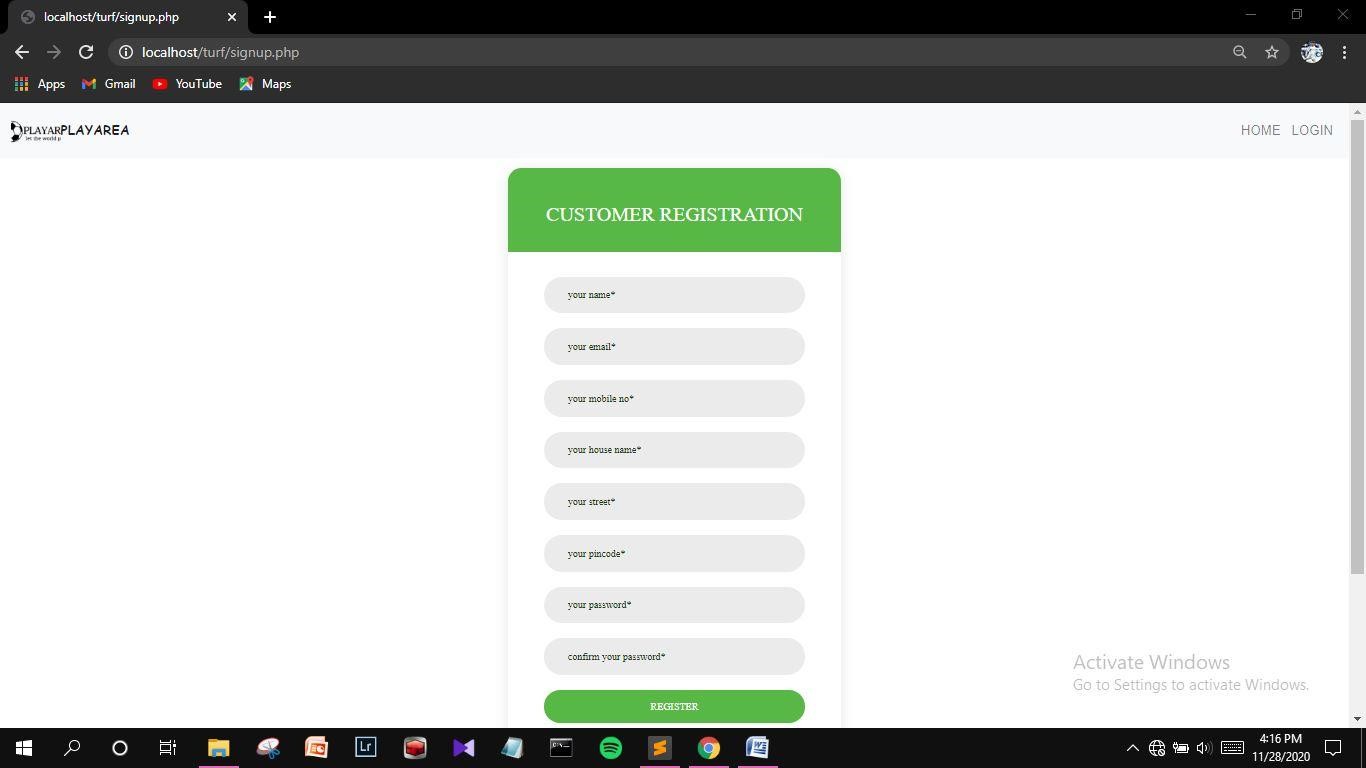
## Login Form

**Description:** This form shows the login page for all the users.



## Customer Registration Form

**Description:** This form shows the registration page for new customers.



**Booking**

**Page**

**Description:**

This is the page where the customer

selects date and

time for booking.

The amount payable for the selected slot is also

displayed in this page.

**Payment**

**Form**

**Description:**

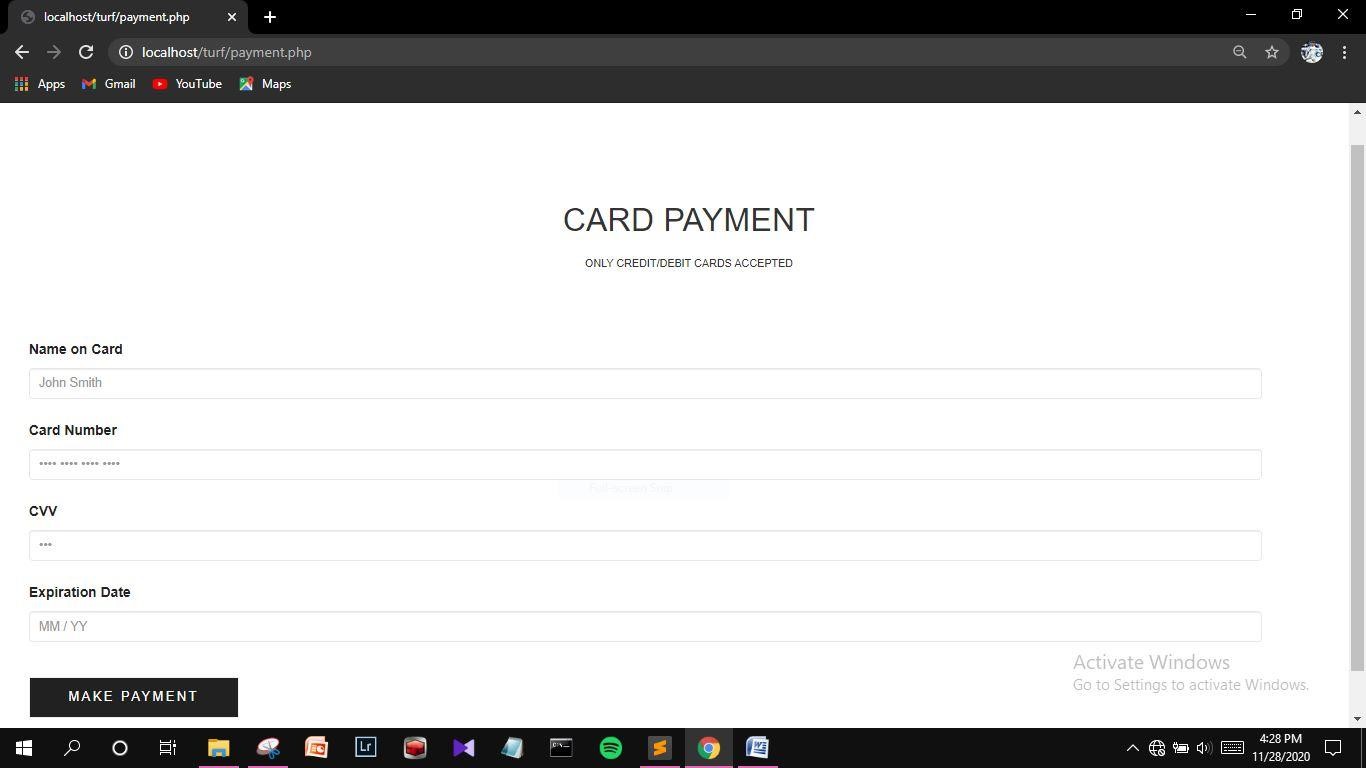
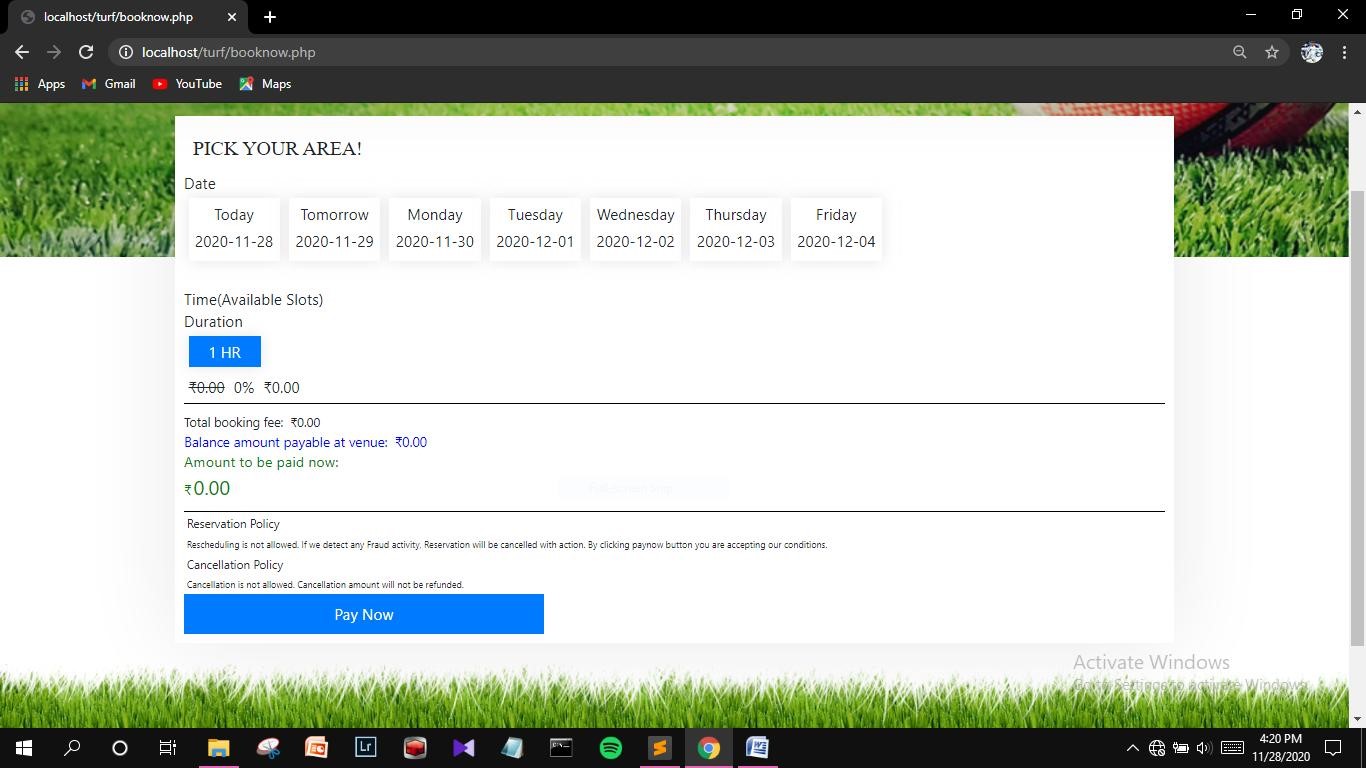
This form

is used

by the customer to make payment to confirm

their booking

.



**Blocking Slots**

**Form**

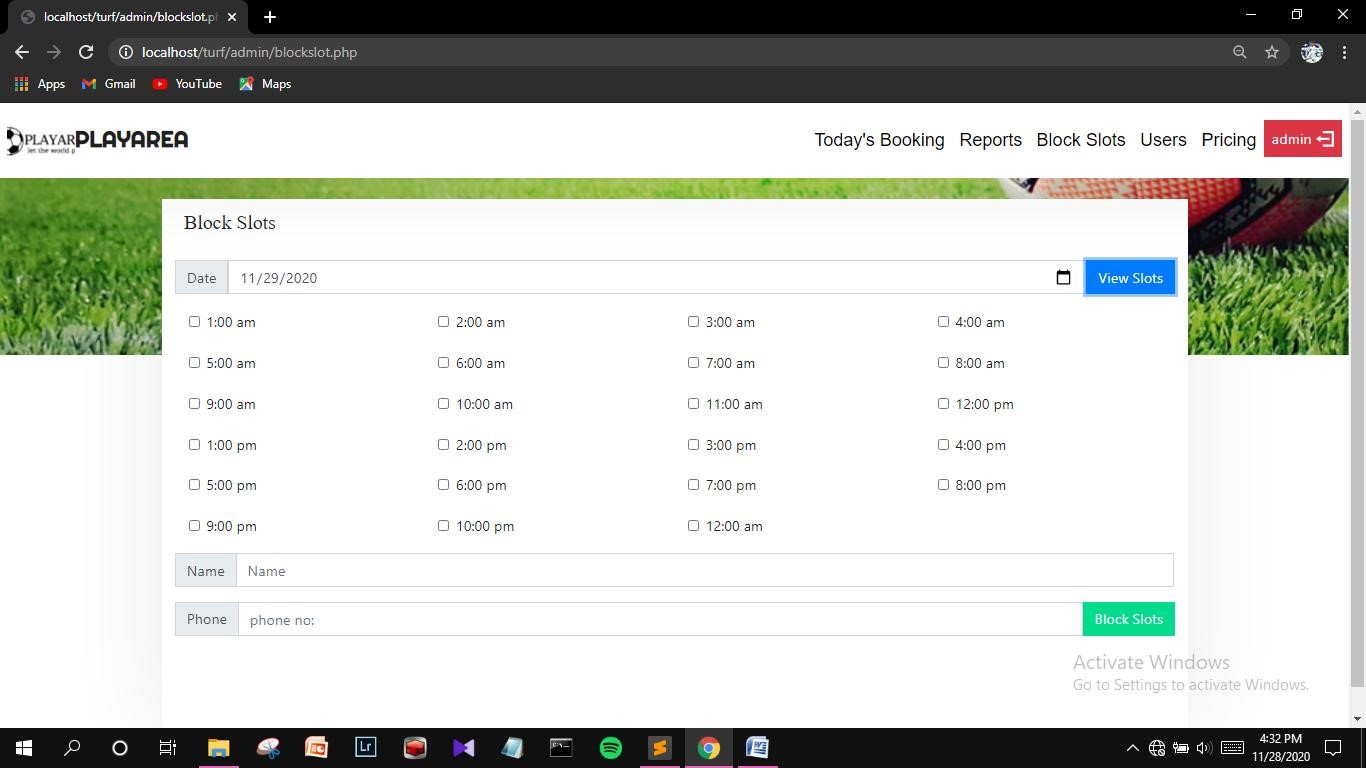
**Description:**

This form

is used to block slots so that the customers cannot

book these slots

.



**Price**

**Updating Page**

**Description:**

This page is used to update prices of the respective timeslots of

the turf

.

**Edit Customer Details**

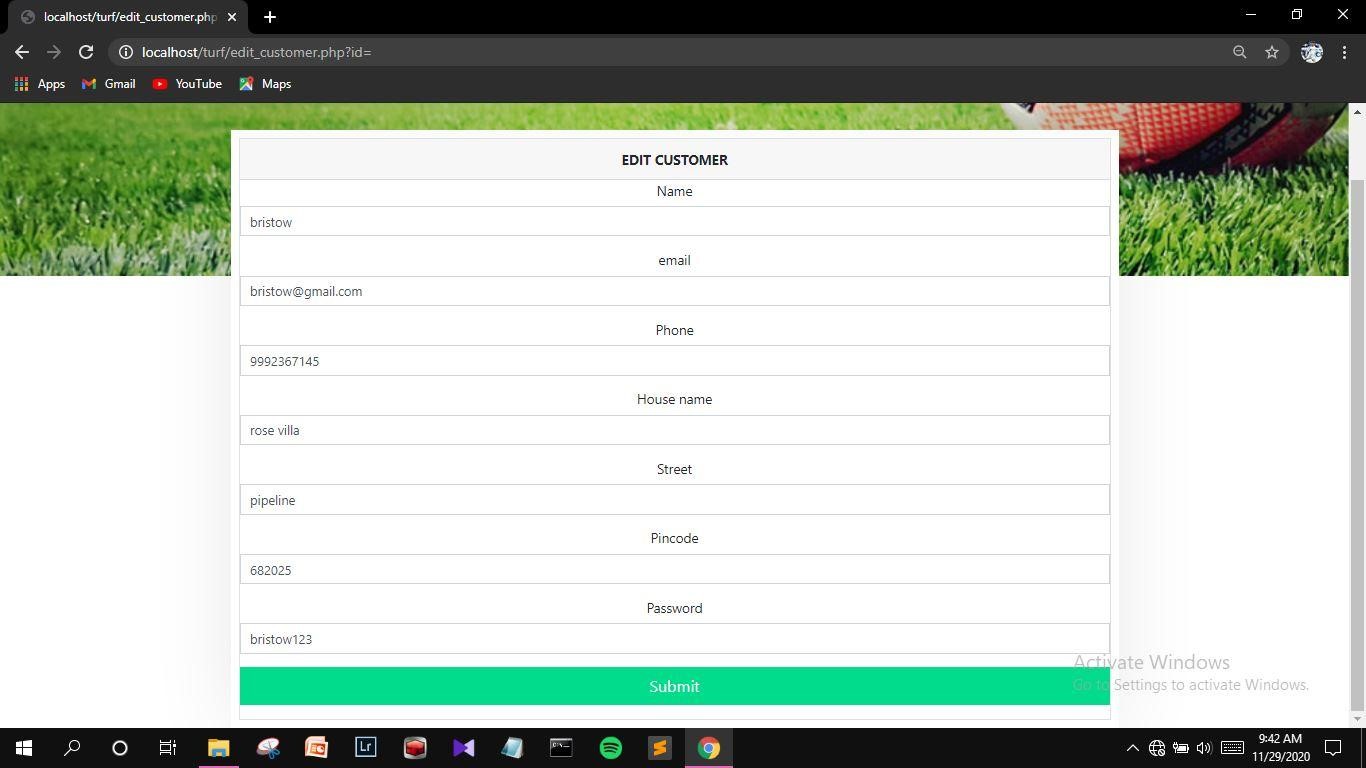
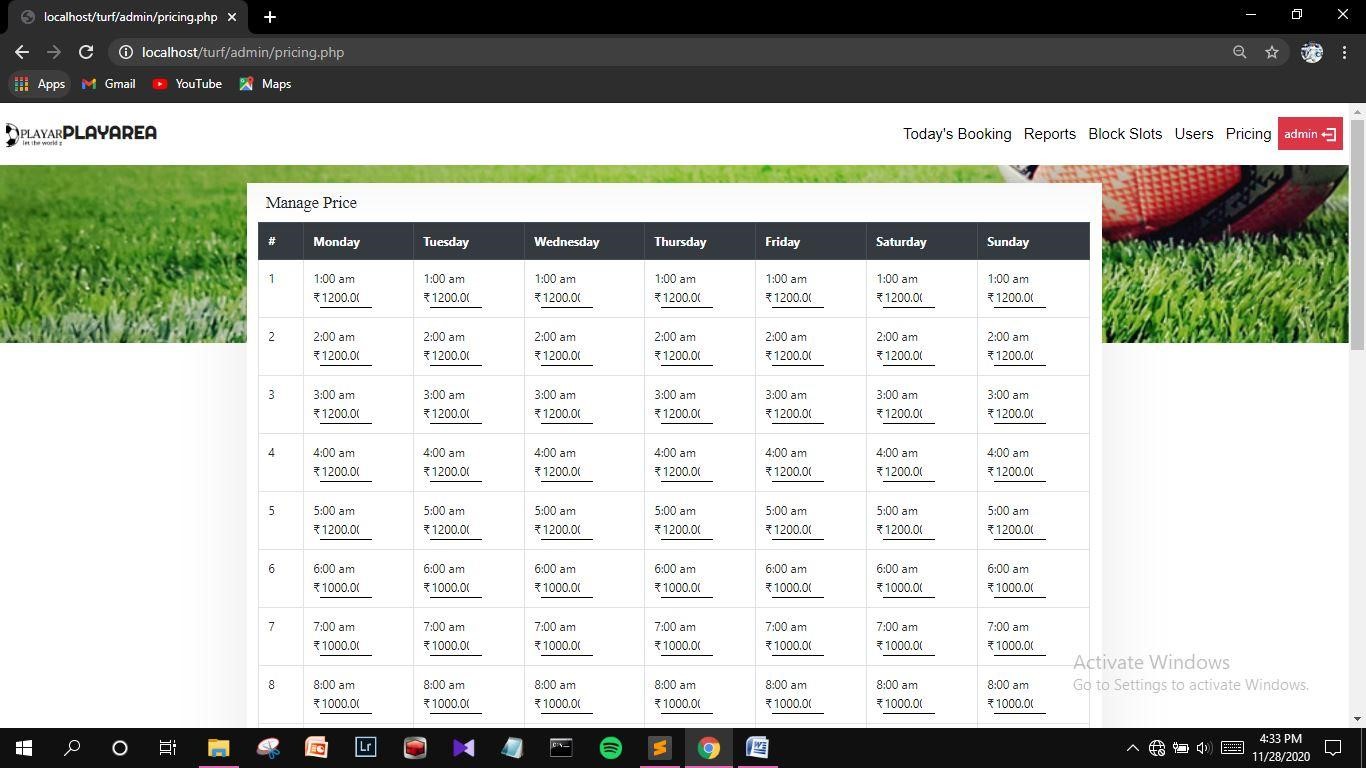
**Form**

**Description:**

This form is used

to update customer details

.



**Booking**

**Report Form**

**Description:**

Th

is form is used get the booking

report of the turf over

particular period of time.

**Day**

**-**

**wise**

**Report Form**

**Description:**

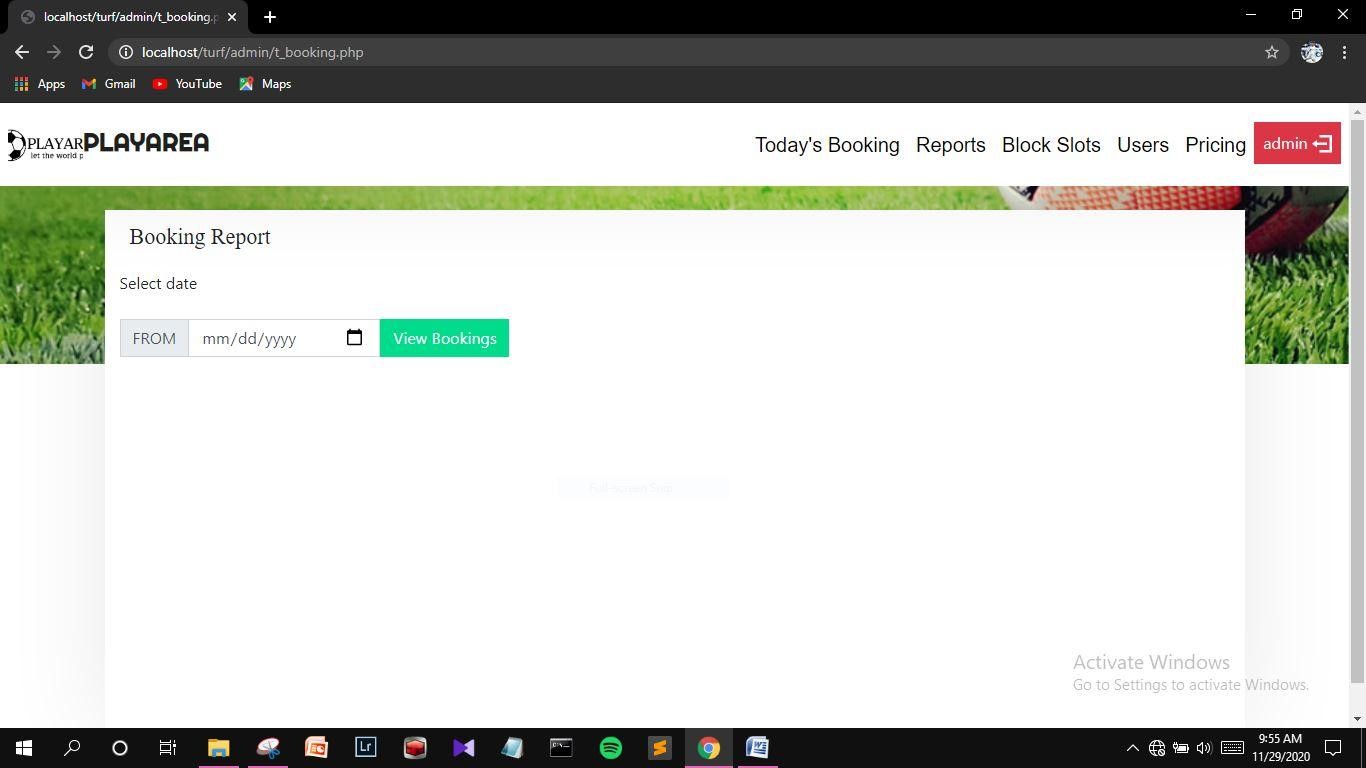
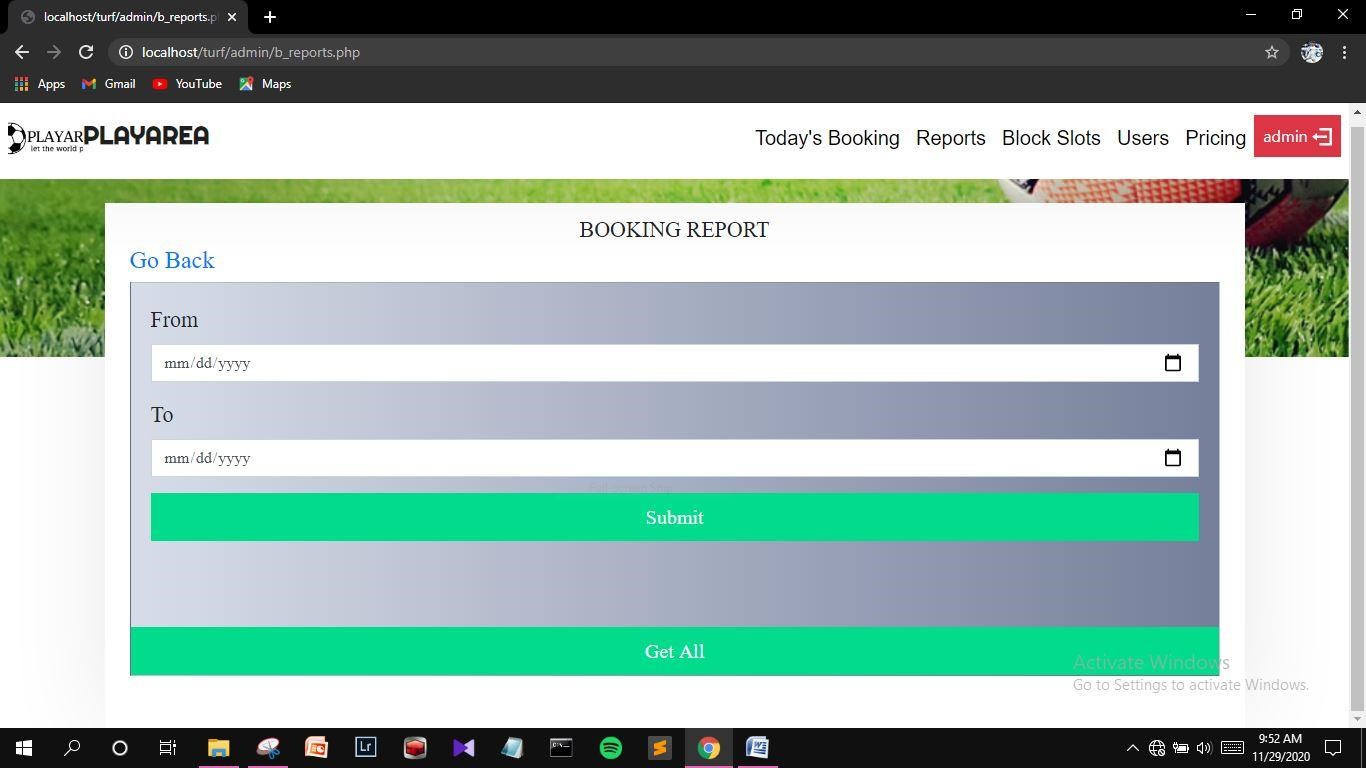
This form is used get the booking

repor

t of the turf on a

particular day

.



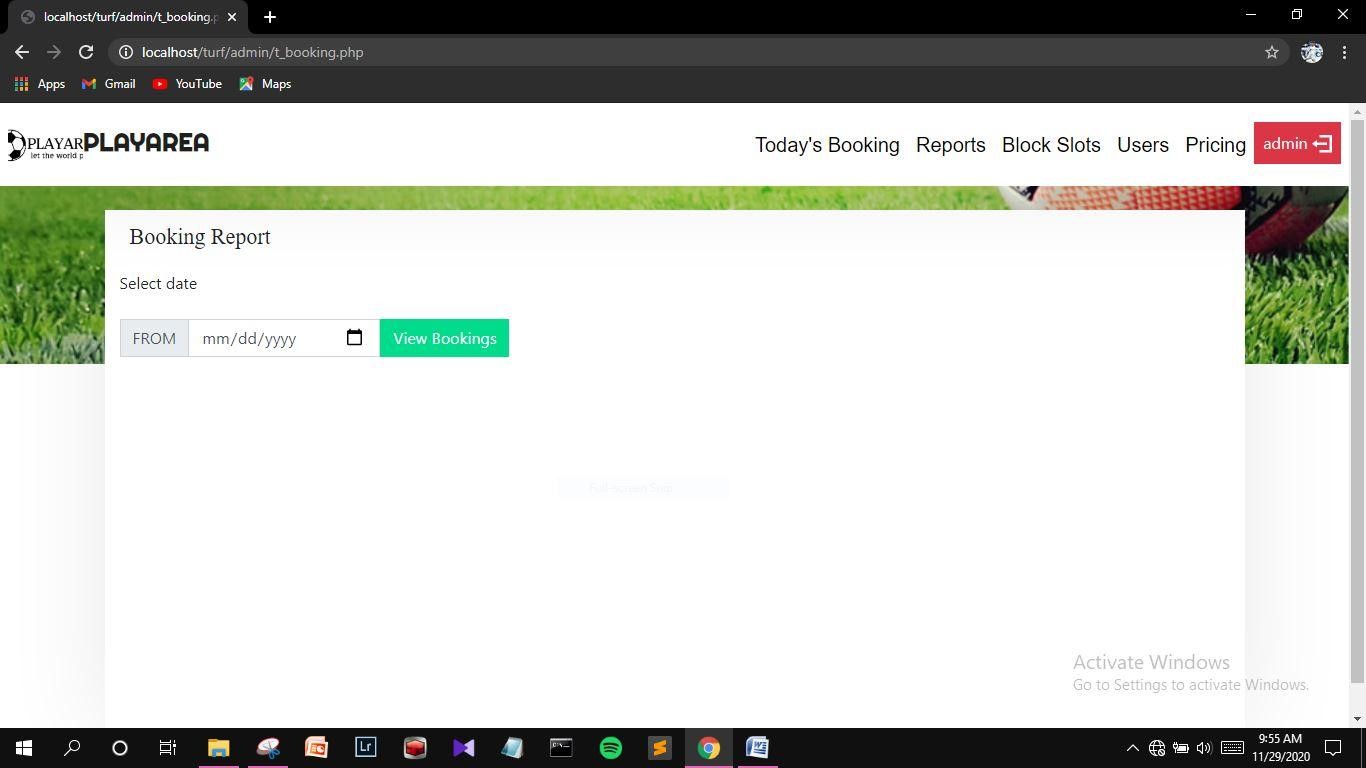
**Finance Report Form**

**Description:**

This form is used get the payment report of the turf over a

particular period of time

.



## 4.2 OUTPUT DESIGN

One of the important features of an information system for users is the output it produces. Output is the information delivered to users through the information system. Without quality output, the entire system appears to be unnecessary that users will avoid using it. Uses generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user through an interactive process, until the result is considered to be satisfactory.

Output design has been an ongoing activity almost from the beginning of the project. In the study phase, outputs were identified and described general in the project directive. A tentative output medium was then selected and sketches made for each output.

In the feasibility analysis, a “best” new system was selected; its description identified the input and output media.

In the design phase the system has included an evaluation and selection of specific equipment for the system. Output design generally deals with the results generated by the system i.e., reports. These reports can be generated from stored or calculated values. Reports are displayed either as screen window preview or printed form. Most end users will not actually operate the information system or enter data through workstation, but they will use the output from the system.

Outputs from computer systems are required primarily to communicate the results of processing to the user. They are also used to provide a permanent copy of these results for later consultation.

**View**

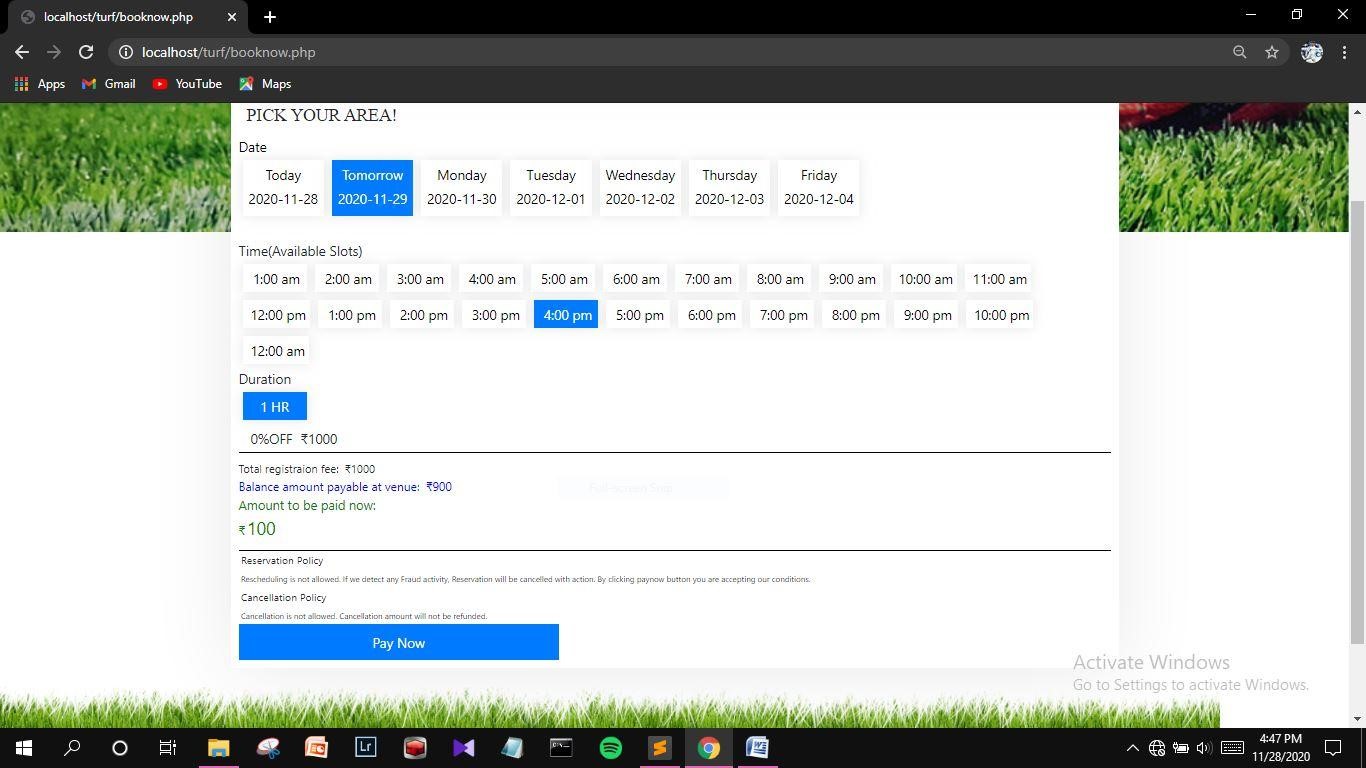
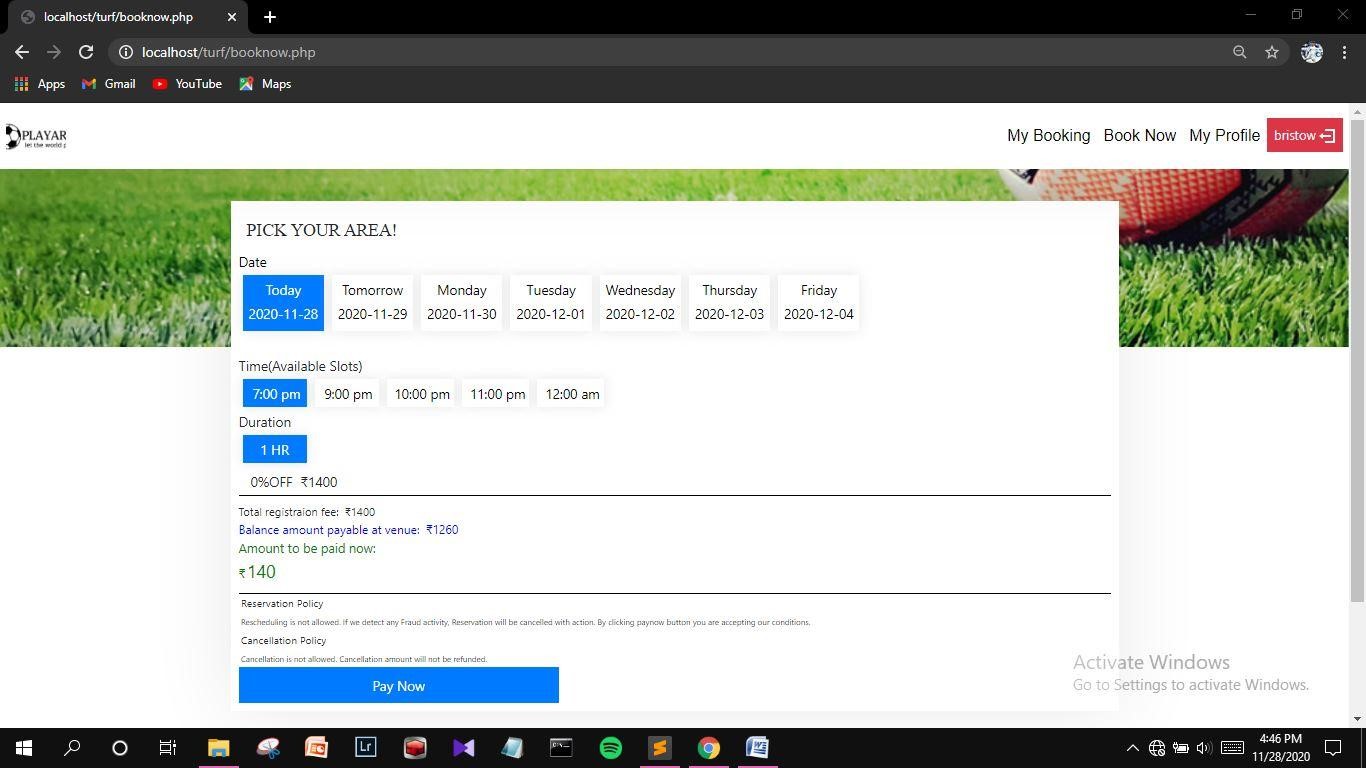
**Active Timeslots**

**Page**

**Description:**

This page shows the slots that are available for booking with

respect to the day selected.



**View**

**Customer Details**

**Page**

**Description:**

This page is used to view and edit

the details of the

customer.

**View**

**Bookings**

**Page**

**Description:**

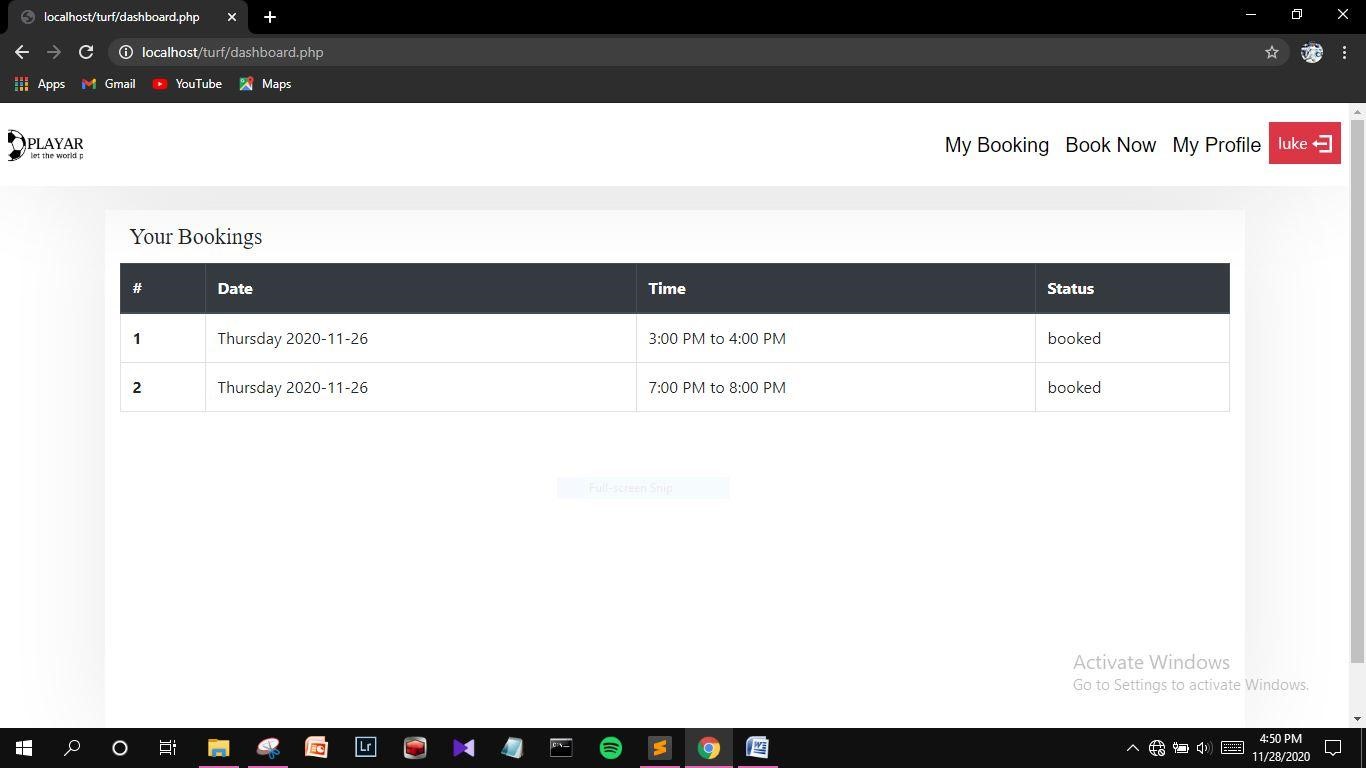
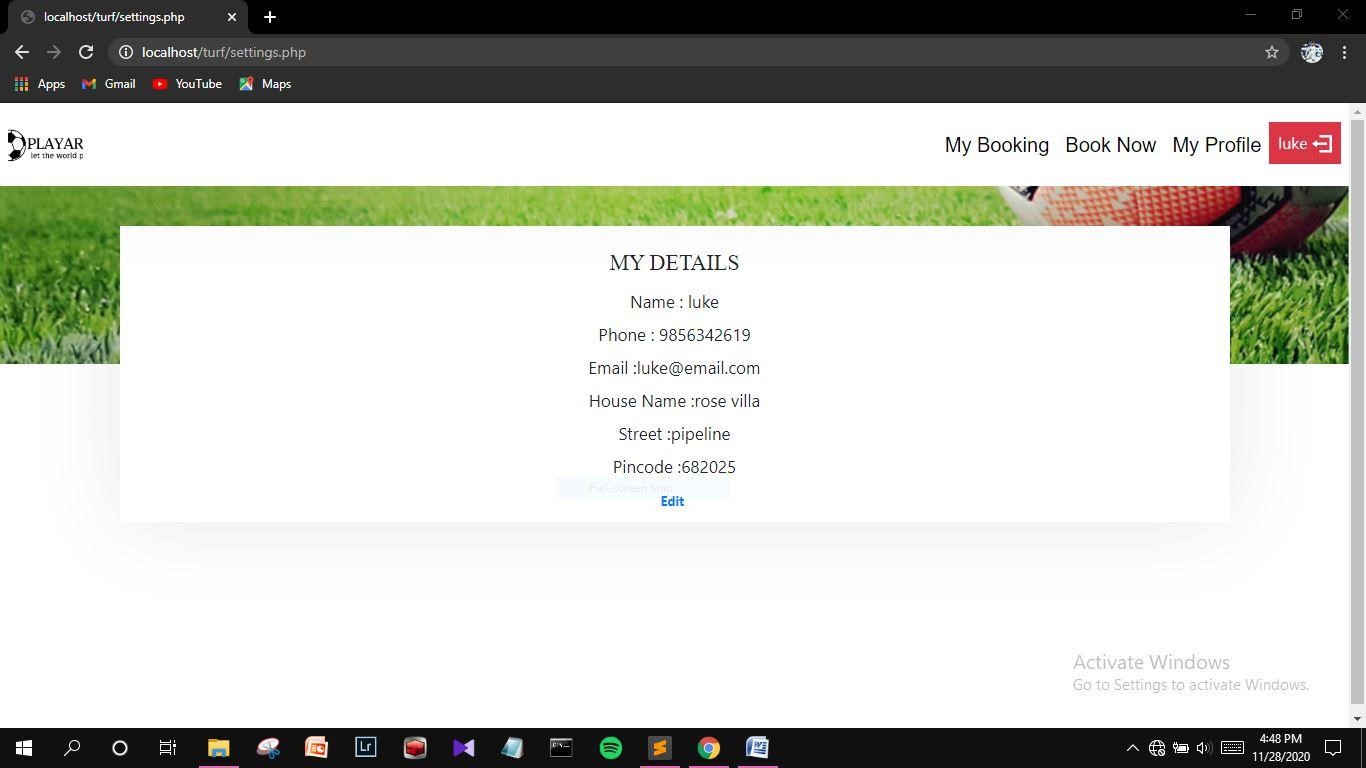
This page shows the details

of

all successful bookings

made by the customer

.



**Payment Success**

**Page**

**Description:**

This page shows th

e confirmation of successful

payment

.

**Present**

**Day’s Bookings**

**Page**

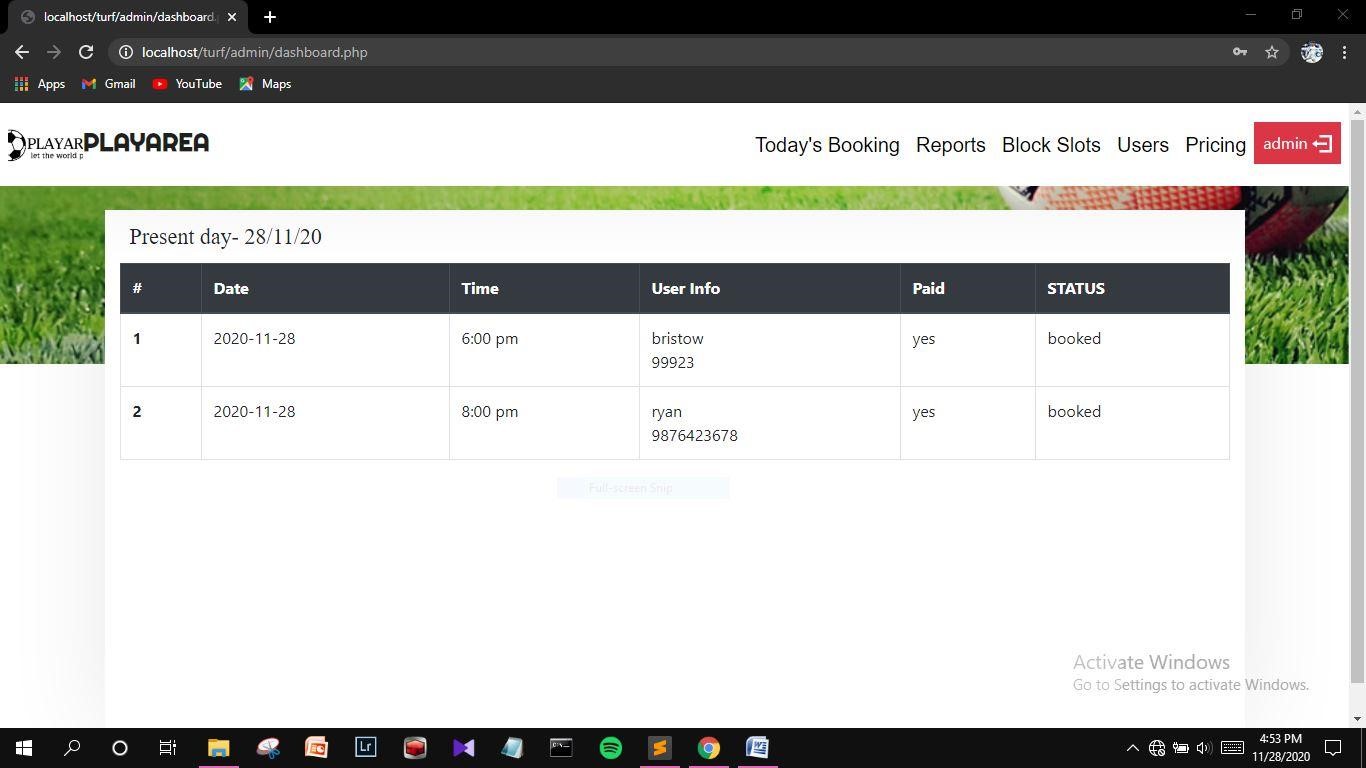
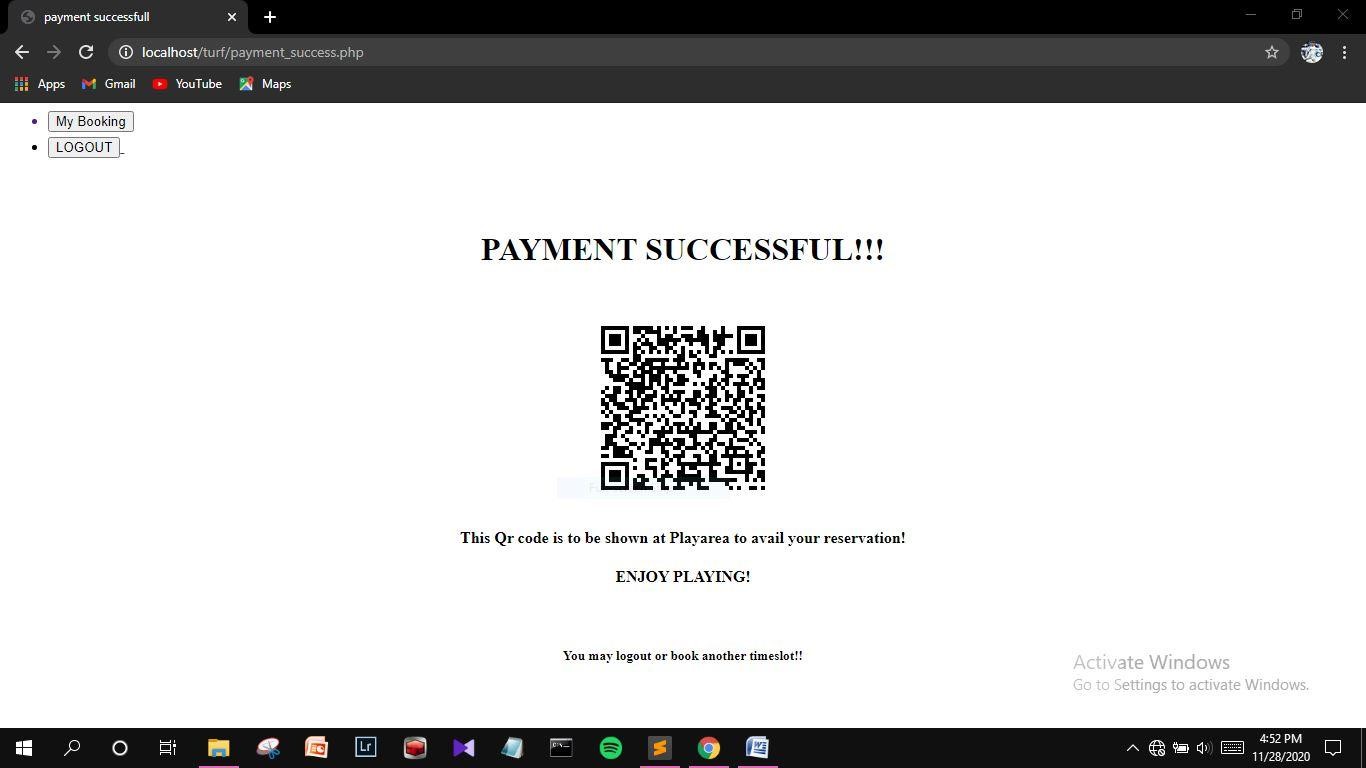
**Description:**

This page shows the details

of all the bookings made for the

present day

.



**View**

**Reports**

**Page**

**Description:**

This page

is used to generate reports according to the

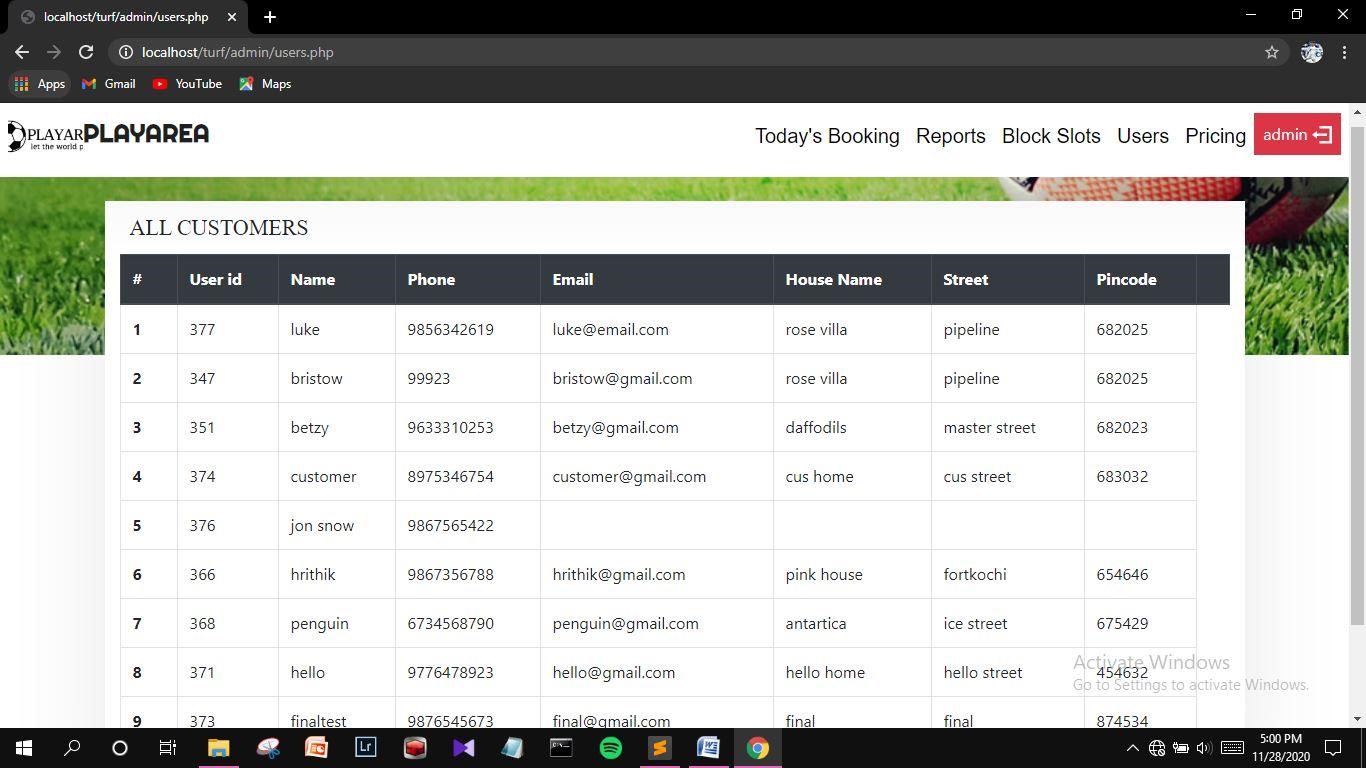
choice of admin.

**View Customer Page**

**Description:**

This page shows the details of r

egistered customers.



**Booking**

**Report**

**Description:**

This page shows the

booking

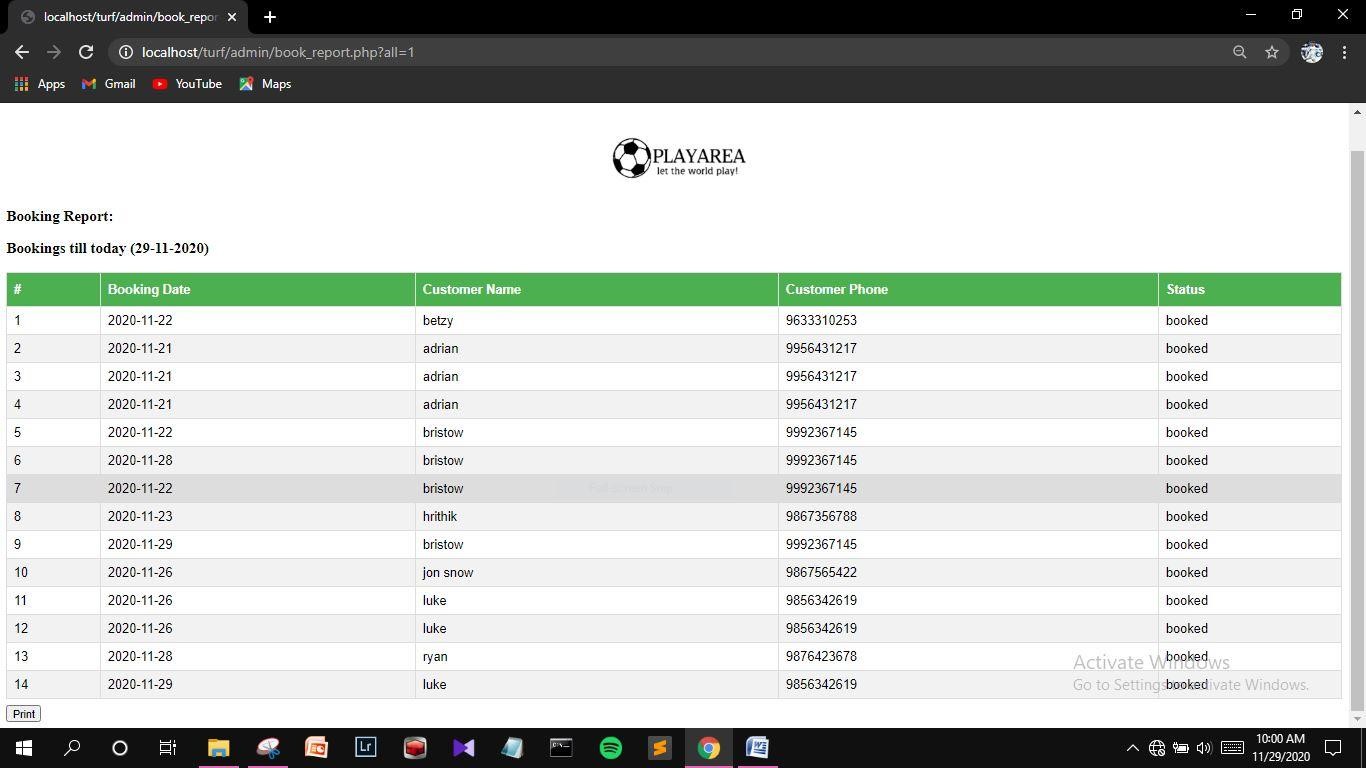
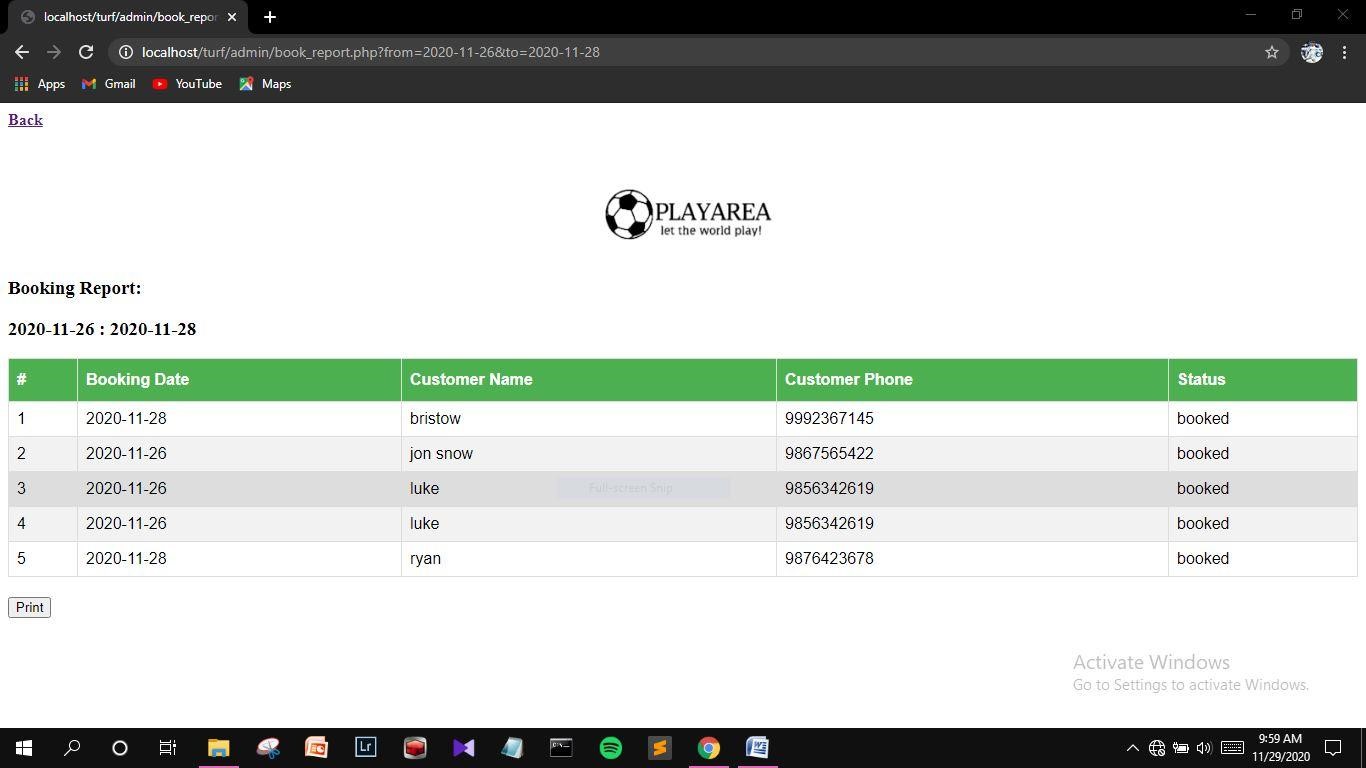
report

over a period of

selected

time

.



**Day**

**-**

**wise Booking Report**

**Description:**

This page shows the

booking

report

of the selected

individual day

.

**Finance**

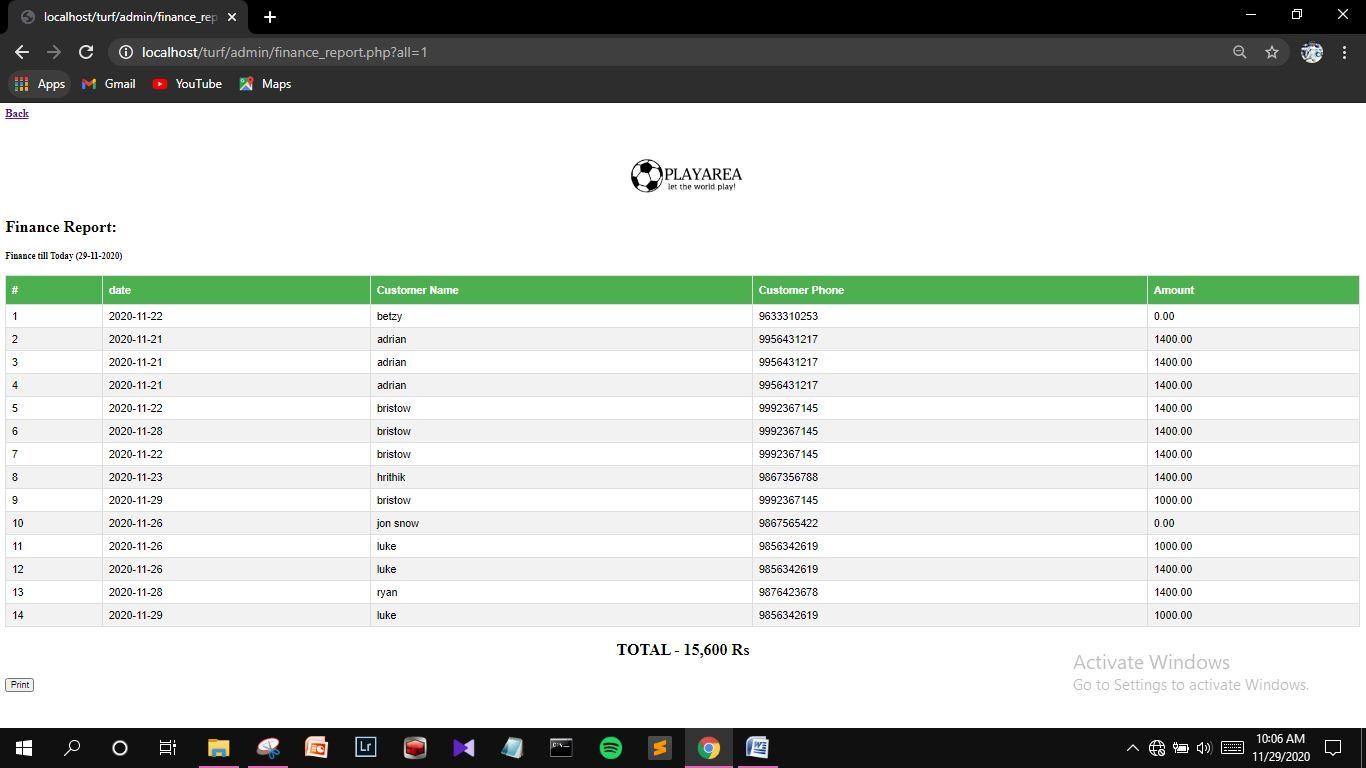
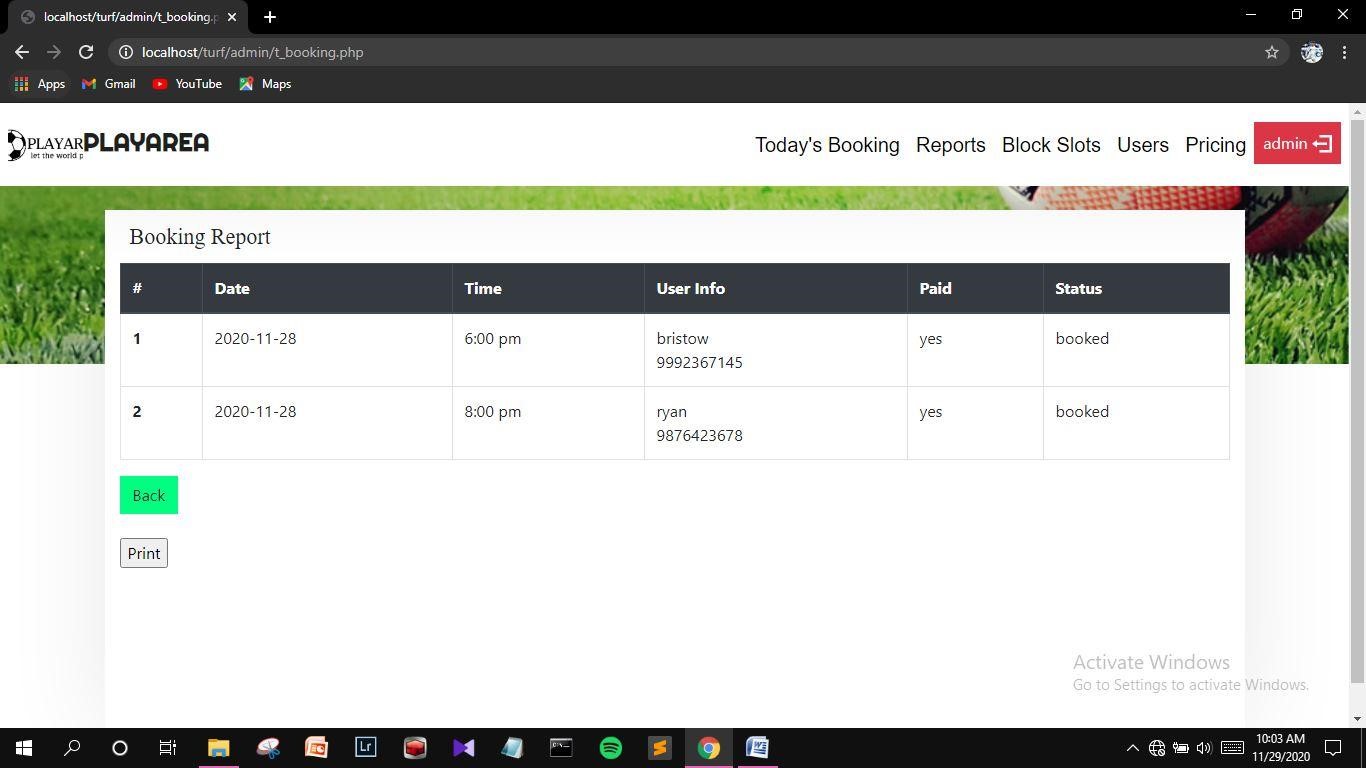
**Report**

**Description:**

This page shows the reports of payments over a selected

period of time

.



## 4.3. DATABASE DESIGN

### 4.3.1 Normalization

Designing a database is a complex task and the normalization theory is a useful aid in this design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form.

A bad database design may lead to certain undesirable situation such us,

* Repetition of information
* Inability to represent certain information  Loss of information

To minimize these anomalies, normalization mat be used. If the database is in a normalization form, the data can be restructured and can maintain it easily. This is important that the database using that we are using may free from data redundancy and inconsistency. For this need we maintain the tables in a normalized manner.

## First Normal Form

A relation is in first normal form (INF), if and only if all its attributes are based on single domain. The objective of normalizing a table is in to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

## Second Normal Form

A table is said to be in Second Normal Form (2NF), when it is in 1NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

## Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Name: tb\_Login**  **Description: stores the login details of the existing users in the system.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **FIELD** |  | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** | | | Userna | me | Varchar(30) | Primary key | Login id | | | Passw | ord | Varchar(20) | Not null | Username | | | User\_t | ype | Varchar(10) | Not null | User type | | |  |  | |  |  |  |   **Table Name: tb\_Customer**  **Description: stores the details of the customer** | | | | | | | | |
|  | | **FIELD** | | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** | |  |
| C\_id | | Varchar(5) | Primary key | Customer id | |
| C\_na | | Varchar(20) | Not null | Customer name | |
| C\_hna | | Varchar(20) | Not null | Customer house name | |
| C\_street | | Varchar(20) | Not null | Customer street | |
| C\_state | | Varchar(10) | Not null | Customer state | |
| C\_pin | | Varchar(8) | Not null | Customer pincode | |
| C\_phno | | Numeric(12) | Not null | Customer mobile number | |
| C\_email | | Varchar(20) | Not null | Customer email | |
|  |  | | | | | |  |
|  | | | | | |
|  | |  | | | |

## Table Name: tb\_timeslots

**Description: stores the details of the timeslots in the system.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** |  | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** | |
| Slot\_id |  | Varchar(5) | Primary key | Time slot id | |
| Slot\_pr | ice | Varchar(5) | Not null | Scheduled slot timings | |
| time |  | Varchar(5) | Not null | Selected time | |
| date |  | Date | Not null | Selected date | |
|  |  | |  |  |  |

## Table Name: tb\_booking

**Description: stores the details of bookings.**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| Book\_id | Varchar(5) | Primary key | Booking id |
| C\_id | Varchar(5) | Foreign key | Customer id |
| Slot\_id | Varchar(5) | Foreign key | Time slot id |
| B\_time | Varchar(5) | Not null | Time of booking |
| B\_date | date | Not null | Date of booking |
| status | Varchar(5) | Not null | Booking status |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Name: tb\_Payment**  **Description: stores the payment details of customers** | | | | | | | |
|  | | **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** | |  |
| Payment\_id | Varchar(20) | Primary key | Payment id | |
| booking\_id | Varchar(20) | Foreign key | Booking id | |
| Pay\_amt | Numeric(10,5) | Not null | Payment amount | |
| Pay\_date | Date | Not null | Payment date | |
| Pay\_status | Varchar(20) | Not null | Payment status | |
|  |  | | | | |  |
| **Table Name: tb\_Cardpay**  **Description:stores the card details of customers**   |  |  |  |  | | --- | --- | --- | --- | | **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** | | Cardpay\_id | Varchar(5) | Primary key | Card payment id | | Cust\_id | Varchar(5) | Foreign key | Customer id | | Pay\_id | Varchar(5) | Foreign key | Payment id | | Card\_no | Varchar(20) | Not null | Card number | | Exp\_date | Varchar(5) | Not null | Expiry of card | | Card\_type | Varchar(10) | Not null | Type of card | | | | | |

## 5.1 INTRODUCTION

Software testing can be looked upon among the many process in organization that provides the last opportunity to correct any plane in the development system. System testing includes selecting tests and test data that have more problem of finding errors. System testing is vital for the success of any software system. The system makes a logical assumption that all part of the system work efficiently and goal is achieved. The system is tested for online response, ability to store and stress recovery from failure and usability. System testing requires a test plan that consists of several key activities and steps for programming and user acceptance testing.

Another benefit of system testing is its utility as a useroriented system before implementation.

## LEVELS OF TESTING

Some of the methods of the system testing are given below.

## Unit testing

In this test each module is tested individually before integration it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing. In this test each module is tested whether it is producing the desired output and if any error occurs it can be corrected easily.

## Integration testing

It is the systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. Thus the relationship between difference modules is checked in this testing for overall performance of testing.

Thus in integration testing step, all errors uncovered are corrected for next testing steps. The objective of the test is to take althea modules such as administrator, user and modules are integrated in this testing step and then the entire program is tested.

## Validation testing

It in where requirements established as a part of software requirements analysis is validated against the software that has been constructed. This test provides the final assurance that the software meets all functional, behavioral and performance requirements. The errors, which are uncovered during integration testing, are connected during this phase.

## Output Testing

No system could be useful if it does not produce the required output in the specific format. Output testing is performed to ensure the correctness of the output and its format. The output generated or displayed by the system is tested asking the user about the format required by them.

## User Acceptance Testing

The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system user at the time of developing. The testing of the software began along with the coding. The unit testing was done for each module in the software. For various inputs such that each line of code is executed at least once.

## 5.2 TEST CASES

A test plan document the strategy that will be used to verify and ensure that a product or system meets its design specification and other requirements. A test plan is usually prepared by or with significant input from test Engineers. Depending on the product and the responsibility of the organization to which the test plan applies.

## Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Form** | **Procedure** | **Expected**  **Result** | **Actual**  **Result** | **Status** |
| Entry Form | Choose whether to Login, view gallery, view services us or view about us |  |  |  |
| Login  Form | Enter valid username and password | Should validate user and provide link to user accounts | Got entry  to accounts | Pass |
| Customer  Form | Enter all mandatory  fields | Should  validate all entered fields and flash a message indicating successful registration | Message  indicating successful registratio n is  shown | Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Booking  Form | Select the required date and time | Should  validate all entered fields and flash a message indicating successful registration | Message  indicating successful booking is shown | Pass |  |
| Payment  Form | Enter all mandatory  fields | Should validate selected date and time. | Message  indicating successful registratio n is  shown | Pass |
| Blocking  slots | Select the required date and time | Should validate selected date | Message  indicating successful blocking is shown | Pass |
| **Integration Testing**   |  |  | | --- | --- | | **Form** | **Expected Result** | | Login and user account forms | Get entry to appropriate user page | | Customer  Form | Must add customer details successfully | | | | |  | | --- | | **Actual Result** | | Appropriate user page is displayed | | Insertion is successful | | | |  | | --- | | **Status** | | Pass | | Pass | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Booking  Form | Must add the specified entry in the database | Specified entry updated | Pass |  |
| Blocking slots Form | Must add the specified entry in the database | Specified entry updated | Pass |
| Payment Must update the Specified entry Pass  Form specified entry in updated  the database    **Validation Testing**   |  |  |  |  | | --- | --- | --- | --- | | **Form** | **Expected Result** | **Actual Result** | **Status** | | Create user | Check all mandatory fields and validate all entered data fields | If any error found display message and the same screen is displayed else record saved and confirmed | Pass | | Edit User | Edit the row corresponding to the value entered | If the value entered is invalid error message is thrown otherwise message indicating successful deletion is flashed | Pass | | | | |

## 6.1 INTRODUCTION

Implementation is that state in the project plan where the theoretical design is put into real test. All the theoretical and practical works are now implemented as a working system. This is the most crucial stage in the life cycle of a project; the project may be accepted or rejected depending on how it gathers confidence among the users. If the user has achieved satisfaction with the new project, then the project can be termed as successful and then onwards its maintenance and other subsequent works can be commenced. The system goes for implementation only after passing through some rigorous testing, especially when it comes to operating system and other system software, the testing and implementation phase assumes greater significance.

The implementation stage involves following tasks

Careful planning.

Investigation of system and constraints.

Design of methods to achieve the change cover.

Evaluation of the changeover method.

## 6.2 INSTALLATION PROCEDURE

Installation of software refers to the final installation of the package in the real environment, to the satisfaction of the intended users and the successful operation of the system. In many organizations, those who commission the software development project will not be the one to operate them. In the initial stage, the person who is not sure that the software will make the jobs easier will doubt about the software. But we have to ensure that the resistance does not build one makes sure that

* The active user must be aware of the benefits of using the system
* Their confidence in the software is built up
* Proper guidance is imparted to the user so that he is comfortable in using the application

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause confusion.

Implementation includes all those activities that take place to convert from the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system into actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification of the system.

## 6.3 IMPLEMENTATION PLAN

Implementation is the most crucial stage in achieving a successful system and for us it is the processing of bringing “Online Turf Booking System” into operational use and training it over to the user. Implementation includes all those activities that take place to convert from the old system to new one.

The basic requirements for implementing the proposed system are already mentioned above. This software provides total security for the operations. That is it prevents any unauthorized access. After successful login the user can go to the form according to the situation.

After completion of the Online Turf Booking System design and coding, the analyst, the user and the management evaluates the system to ensure that it fulfil all its goals. Thus with the implementation of the project the critical design is turned into a working system. System implementation plan is concerned with writing program, creating databases, testing programs and operational plans.

## 7.1 FUTURE ENHANCEMENT

The system has been developed with flexibility in mind. The requirement of the company is bound to change as and when new operations are included. Keeping in view advancements that are being made in technology it is necessary that the system be able to cope up with the changes that are bound to happen.

So in today's world of mobile technology the software “Online Turf Booking

System” if integrated with the mobile will be an added advantage. The mobile users will get instant alerts from this site. The software if we create a mobile app or an alert system for more interaction with the user and also widening the reach of the system to its users.

The system entitled “Online Turf Booking System” provides maximum user interaction and flexibility. The system users stored procedures on the database. This also can be enhanced in the future.

# BIBLIOGRAPHY

Website References

* www.stackoverflow.com
* https://stackoverflow.com/en-us/library/aa228849(v=vs.60).aspx
* [www.coderewind.com](http://www.coderewind.com/) -
* [www.studentsprojectsguide.com](http://www.studentsprojectsguide.com/) -
* http://studentsprojectsguide.in/15698264-2?show=86
* [www.dreamincode.net–](http://www.dreamincode.net/)
* [http://www.dreamincode.net/forums/topic/132258-editing-msflexgridby-users/](http://www.dreamincode.net/forums/topic/132258-editing-msflexgrid-by-users/)
* [www.w3schools.com](http://www.w3schools.com/) -

# APPENDIX A

//customer or admin login

<?php session\_start(); include('include/db.php'); $email = $\_POST['user'];

$password = $\_POST['pass'];

//to prevent from mysqli injection

$email = stripcslashes($email);

$password = stripcslashes($password);

$email = mysqli\_real\_escape\_string($conn, $email);

$password = mysqli\_real\_escape\_string($conn, $password);

$sql = "select phone,type from user where email = '$email' and password =

'$password'";

$result = mysqli\_query($conn, $sql);

$row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

$count = mysqli\_num\_rows($result); if($row['type']=="admin"){ header("Location:admin/dashboard.php");

}else{

header("Location:dashboard.php");

}

} else{

echo '<script>alert("Login failed. Invalid username or password")</script>';

}

?>

<form name="f1" action = "authentication.php" onsubmit = "return true" method = "POST" class="login100-form validate-form p-l-55 p-r-55 p-t-

178">

<span class="login100-form-title"> LOGIN HERE!

</span>

<div id = "frm">

<center><p>

<div class="wrap-input100 validate-input m-b-16" data-validate="Please enter username">

<label></label>

<input type = "text" id ="user" name = "user" placeholder="your email id\*" class="input100" />

<span class="focus-input100"></span>

</div>

</p>

<p>

<div class="wrap-input100 validate-input" data-validate = "Please enter password"> <label></label>

<input type = "password" id ="pass" name = "pass"

placeholder="password\*" class="input100" />

<span class="focus-input100"></span>

</div>

</p>

<p>

<br>

<div class="container-login100-form-btn">

<input type = "submit" id = "btn" value = "login" class="login100-form-btn" />

</div>

<div class="flex-col-c p-t-170 p-b-40">

<span class="txt1 p-b-9">

Don’t have an account?

</span>

<a href="signup.php" class="txt3">

Sign up now

</a>

</div>

</p>

</form>

<form name="f1" action = "authentication\_signup.php" onsubmit = "return true" method = "POST" class="login100-form validate-form p-l-55 p-r-55 pt-178">

<span class="login100-form-title">

CUSTOMER REGISTRATION

</span>

<center>

<p>

<label></label>

<input type = "text" id ="name" name = "name" placeholder="your name\*" class="input100" pattern="[A-Za-z]{1,32}" title="Name can only contain

Alphabets" required="" />

</p>

<p>

<label></label>

<input type = "text" id ="user" name = "user" placeholder="your email\*" class="input100" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$" title="Enter valid email" required="" />

</p>

<p>

<label></label>

<input type = "text" id ="phone" name = "phone" placeholder="your mobile no\*" class="input100" pattern="[6-9]{1}[0-9]{9}" title="Enter valid mobile number" required="" />

</p>

<p>

<label></label>

<input type = "text" id ="hna" name = "hna" placeholder="your house name\*" class="input100" required="" />

</p>

<p>

<label></label>

<input type = "text" id ="street" name = "street" placeholder="your street\*" class="input100" required="" />

</p>

<p>

<label></label>

<input type = "text" id ="pin" name = "pin" placeholder="your pincode\*" class="input100" required="" />

</p>

<p>

<label></label>

<input type = "password" id ="pass" name = "pass" placeholder="your password\*" class="input100" required="" />

</p>

<p>

<label></label>

<input type = "password" id ="password2" name = "password2" placeholder="confirm your password\*" class="input100" required="" />

</p>

<p>

<br>

<div class="container-login100-form-btn">

<input type = "submit" id = "btn" value = "REGISTER" class="login100form-btn" />

</div>

<div class="flex-col-c p-t-170 p-b-40">

<a href="login.php" class="txt3">

Already have an account?-Login!

</a>

</div>

</p>

</center>

</form>

//customer registration

<?php

include('include/db.php'); $email = $\_POST['user'];

$password = $\_POST['pass'];

$phone = $\_POST['phone'];

$name = $\_POST['name'];

$hna = $\_POST['hna'];

$street = $\_POST['street'];

$pin = $\_POST['pin'];

//to prevent from mysqli injection

$email = stripcslashes($email);

$name = stripcslashes($name);

$phone = stripcslashes($phone);

$password = stripcslashes($password);

$hna = stripcslashes($hna);

$street = stripcslashes($street);

$pin = stripcslashes($pin);

$email = mysqli\_real\_escape\_string($conn, $email);

$password = mysqli\_real\_escape\_string($conn, $password);

$name = mysqli\_real\_escape\_string($conn, $name);

$phone = mysqli\_real\_escape\_string($conn, $phone);

$hna = mysqli\_real\_escape\_string($conn, $hna);

$street = mysqli\_real\_escape\_string($conn, $street);

$pin = mysqli\_real\_escape\_string($conn, $pin);

$sql = "select \* from user where email = '$email'";

$result = mysqli\_query($conn, $sql);

$row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

$count = mysqli\_num\_rows($row);

if($count == 1)

{

echo "User already exist";

} else{

mysqli\_query($conn,"insertintouser (name,email,phone,password,hna,street,pin) values('$name','$email','$phone','$password','$hna','$street','$pin')"); header("Location:login.php");

}

?>

//edit customer details <?php include\_once("include/nav.php");

include\_once("include/db.php");

$name='';

$phone='';

$email='';

$hna='';

$street='';

$pin=''; $msg=''; if(isset($\_GET['id']))

{

$id=get\_safe\_value($con,$\_GET['id']); echo $id;

$res = mysqli\_query($con, "select \* from user where id='$id'");

$check=mysqli\_num\_rows($res); if($check>0){

$row=mysqli\_fetch\_assoc($res);

$name=$user\_data['name'];

$phone=$user\_data['phone'];

$email=$user\_data['email'];

$hna=$user\_data['hna'];

$street=$user\_data['street'];

$pin=$user\_data['pin'];

}else{ header('location:settings.php');

die();

}

} if(isset($\_POST['submit']))

{

$name=get\_safe\_value($con,$\_POST['name']);

$email=get\_safe\_value($con,$\_POST['email']);

$phone=get\_safe\_value($con,$\_POST['phone']);

$hna=get\_safe\_value($con,$\_POST['hna']);

$street=get\_safe\_value($con,$\_POST['street']);

$pin=get\_safe\_value($con,$user\_data['pin']);

$password=get\_safe\_value($con,$\_POST['password']);

$res=mysqli\_query($con,"select \* from user where id='$id'");

$check=mysqli\_num\_rows($res); if($check>0){

if(isset($user\_data['id']) && $user\_data['id']!='')

{

$getData=mysqli\_fetch\_assoc($res); if($id== $user\_data['id'])

{

}

else

{

$msg="customer already exists";

}

}else{

$msg="customer already exists";

}

}

if($msg=='')

if($msg=='')

{

if(isset($\_GET['id']) && $\_GET['id']!='')

{

$sql="update user set='$name',phone='$phone',email='$email',hna='$hna',street='$street',pin='$ pin',password='$password' where id='$id'"; mysqli\_query($con,$sql);

}

else

{

header('location:settings.php');

}

} echo $id;

?>

**APPENDIX B**

## Acronyms

SQL - Structured Query Language

DFD - Data Flow Diagram

ERD - Entity Relationship Diagram

IDE - Integrated Development Environment OS - Operating System