|  |  |
| --- | --- |
|  | **Cognizant Academy** **DREAM Theme Park Ticket Booking Automation** **HTML Technologies Knock Out Challenge**  **Version 1.0** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** |  |  |  | | **Role** |  |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc23440118)

[1.1 Purpose 3](#_Toc23440119)

[1.2 Definitions & Acronyms 3](#_Toc23440120)

[1.3 Project Overview 3](#_Toc23440121)

[1.4 Scope 3](#_Toc23440122)

[1.5 Target Audience 4](#_Toc23440123)

[1.6 Hardware and Software Requirements 4](#_Toc23440124)

[1.6.1 Hardware Requirements 4](#_Toc23440125)

[1.6.2 Software Requirements 4](#_Toc23440126)

[2.0 Functional Requirements 5](#_Toc23440127)

[2.1 Functional Requirements 5](#_Toc23440128)

[3.0 Design Specification 7](#_Toc23440131)

[3.1 UI Design 7](#_Toc23440134)

[3.2 General Design Constraints](#_Toc23440135) 8

[4.0 Submission 9](#_Toc23440138)

[4.1 Code submission instructions 9](#_Toc23440139)

[5.0 Change Log 10](#_Toc23440140)

# Introduction

## Purpose of this document

Dream Theme park is a famous amusement park that attracts lot of people everyday. The queue outside the ticket counter is usually long and people wait long to get their tickets.So in order to help their visitors save on their time and energy, the theme park wants to automate its booking process so that the visitors can book the tickets online from the official website. All that the visitors need to do is log on to the website and provide their details.

Dream Theme Park Automation wants to design a web page for the ticket booking that requires the visitors to provide their details. Create a Web page “Book Your Tickets” that has the following fields such as name, address, email, mobile number along with the location of the theme park, date of visit and number of people visiting the theme park.  Also booking type needs to be entered.  It can be either Normal or School or Corporate. The styles have to be applied using CSS3 and the Ticket amount validated and displayed in Javascript.

1.2 Definitions & Acronyms

|  |  |
| --- | --- |
| Definition / Acronym | Description |
| Nill |  |
|  |  |
|  |  |
|  |  |

## Project Overview

This project enables to implement the concepts of Web technologies such as HTML5, CSS3 and Javascript, and helps to acquire the skills required by a Web designer. The HTML5, CSS3 and Javascript requirements are given precisely so that the associate can implement the various concepts learned in all these web technologies,The individual associate is expected to carry out the knock out challenge and complete it within 2 hours.

## Scope

The scope of the Dream Theme Park Automation is focused on the following

1. Designing a Web page “Book Your Tickets” with the fields specified.
2. Applying the styles using CSS to the components.
3. Using JavaScript for calculating the total amount of tickets.

## Target Audience

Learner Level

## Hardware and Software Requirements

### Hardware Requirements

|  |  |  |
| --- | --- | --- |
| # | Item | Specification/Version |
|  |  |  |
|  |  |  |
|  |  |  |

### Software Requirements

|  |  |  |
| --- | --- | --- |
| # | Item | Specification/Version |
| 1. | HTML | 5 |
| 2. | CSS | 3 |
| 3. | Javascript |  |

Note: All the required hardware and software will be provided in the TekStac platform

# Functional Requirements

## Functional Requirements

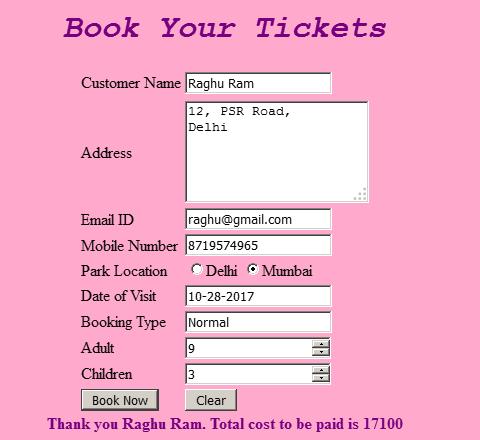
|  |  |  |
| --- | --- | --- |
| Req. # | Req. Name | Req. Description |
| 1 | Design a Web page “Book Your Tickets” with the specified fields. | |  |  |  |  | | --- | --- | --- | --- | | **Label Name** | **Component  Name** | **Component Id** | **Description** | | Customer Name | customerName | customerName | To enter the name of the customer.  **It should be mandatory.** | | Address | address | Address | To enter the address of the customer.  **Provide the correct tag and attribute to create a text area with  5 rows and 20 columns.**  **It should be mandatory.** | | Email ID | customerEmail | customerEmail | To enter the email of the customer.  Use type=”email” or type=”text. If you use type=”text”,use “pattern” attribute along with it.  **It should be mandatory.** | | Mobile Number | customerMobileNumber | customerMobileNumber | To enter the mobile number.  Use type=”tel” or type=”text or type=”number”. If you use type=”text” or type=”number”use “pattern” attribute to accept whether the 10 digit mobile no starts with either one of the following numbers:  6,7,8 or 9  **It should be mandatory.** | | Park Location | location | locationMumbai  locationDelhi | Select the Park location. Use two locations Mumbai and Delhi.  **Provide the correct input type to make this component a radio button.**  **It should be mandatory.** | | Date of Visit | dateOfVisit | dateOfVisit | To select the date of visit.  Use type=”date” or type=”text. If you use type=”text”,use “pattern” attribute to accept the dateOfVisit in the following format :  mm/dd/yyyy  **It should be mandatory.** | | Booking Type | bookingType | bookingType | An auto-complete feature should be available to the user for the following options.  **Normal, School, Corporate**  **(Use DataList “bookType” to make use of auto-complete feature with option Normal,School,Corporate).**  **It should be mandatory.** | | Adult | adults | adults | To enter the number of adults.  Provide the input type so that this field accepts only digits.  **Set minimum value as 0 and maximum value as 500. It should be mandatory.** | | Children | children | children | To enter the number of children.  Provide the input type so that this field accepts only digits.  **It should be mandatory.** | | Book Now | submit | submit | The input type submit must be used. | | Clear | reset | reset | Reset Button.  On clicking this button, all fields should be reset.  Provide the input type for this component. |    NOTE: The text highlighted in bold in the Description needs to be implemented in the code to complete the web page design. Use the Label Name, Component Name and Component Id as given  All validations should be based on HTML 5 (Do not use Javascript). All the fields are mandatory |
| 2 | Apply the mentioned styles using CSS to the components. | 1.       Body color should be “#FFAACC”.  2.       The heading should be done using <h1> tag, the text color should be “#770080”, font should be “Courier New”, style should be “italic” and it should be aligned to the center of the webpage.  3.       The result should be bold and color of the text should be #770080  Note: Fill in the  style tag so that the given styles are applied to the components.  (Do not use Inline CSS) |
| 3 | Use JavaScript for calculating the total amount of tickets | Use JavaScript for doing the following calculation:  Customer on entering the valid values and clicking the submit button the total amount should be calculated based on the assumptions given:   * Ticket cost per Adult : Rs. 1500 * Ticket cost per Child : Rs. 1200 * If Booking Type is Normal, no discount is provided. * If Booking Type is School, 50% discount is provided on the total ticket cost * If Booking Type is Corporate, 25% discount is provided on the total ticket cost   Hence Total cost = Total Ticket Cost  -  discount  **The result should exactly match with the following message:**   “**Thank you** *customer name***. Total cost to be paid is** *total cost***”  in a div tag  with   id  “result**”.  **Note: Use getElementsByName or getElementById function to retrieve the values**  **Substitute** *customer name* with the value of the customerName field.  **Substitute** *total cost* with the calculated total cost. |
| 4 | form Tag with attribute onsubmit | **form** Tag is already given in the code template. Do not change the code template and do the coding as per requirement and specification.  Make sure that the onsubmit attribute in the form tag is invoking the JavaScript function like "**return calculateCharges()**".  Also ensure that “return false;” statement is the last line of the JavaScript function **“calculateCharges()”.** |

# Design Specification

## UI Design

* + - 1. The Sample web page should look as follows



* + - 1. And on providing the values the web page should look as follows :

## General Design Constraints

* + - 1. The label name and the component name should be correctly specified as given in the document. Validation of all the fields should be based on HTML 5 (Do not use Javascript). All the fields are mandatory.
      2. For the Styles, do not use Inline CSS.
      3. For Javascript, use getElementsByName or getElementById function to retrieve the values

# Submission

## Code submission instructions

Do not change the code skeleton given as your code will be auto evaluated.

You can validate your solution against sample test cases during the assessment duration.

Your last submitted solution will be considered for detailed evaluation.

Make sure to submit the solution before the specified time limit. Once the time is up, you will not be allowed to submit the solution.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title></title>

<meta name="description" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="">

<style>

#costtable{

font-size: 15px;

text-align: center;

float: right;

border-style: solid;

}

h1

{

color: #08088A;

text-align: center;

font-family: verdana;

}

body

{

background-color: #CEF6CE;

font-family: Verdana;

}

</style>

</head>

<body>

<h1>Air Cargo Service</h1>

<form onsubmit="return false">

<table>

<tr>

<td>

<label for="accnumber">Customer Logistics Account Number</label>

</td>

<td>

<input type="text" name="accnumber" id="accnumber" placeholder="Enter the Unique ID" pattern="[a-zA-z]{4}[\_][0-9]{3}" required="required">

</td>

</tr>

<tr>

<td>

<label for="mobile">Mobile</label>

</td>

<td>

<input type="text" name="mobile" id="mobile" required="required" maxlength="10" pattern="[789][0-9]{9}">

</td>

</tr>

<tr>

<td>

<label for="emailid">Email-ID</label>

</td>

<td>

<input type="text" name="emailid" id="emailid" required="required">

</td>

</tr>

<tr>

<td>

<label for="item" id="category">Item to be Shipped</label>

</td>

<td>

<input type="text" name="item" id="item" placeholder="Item Name" required="required" pattern="[a-zA-Z\s]+">

</td>

</tr>

<tr>

<td>

<label for="itemtype">Item Category</label>

</td>

<td>

<select required="required" id="category" name="itemtype">

<option id="category" value="Electronics">Electronics </option>

<option id="category" value="Attires">Attires</option>

<option id="category" value="Toys">Toys</option>

<option id="category" value="Non Poisonous Food">Non Poisonous Food</option>

</select>

</td>

</tr>

<tr>

<td>

<label for="address">Shipment Address</label>

</td>

<td>

<textarea name="address" id="address" required="required"></textarea>

</td>

</tr>

<tr>

<td>

<label for="location">Location</label>

</td>

<td>

<select required="required" name="location" id="location">

<option id="category1" value="London - 1800" class ="category">London - 1800</option>

<option id="category2" value="Singapore - 1400" class ="category">Singapore - 1400</option>

<option id="category3" value="New York - 3100" class ="category">New York - 3100</option>

<option id="category4" value="Mexico City - 3300" class ="category">Mexico City - 3300</option>

<option id="category5" value="Tokyo - 2800" class ="category">Tokyo - 2800</option>

</select>

</td>

</tr>

<tr>

<td>

<label for="weight" name="weight">Weight in kilogram</label>

</td>

<td>

<input type="number" name="weight" id="kilogramval" min="5" max="36" required="required">

</td>

</tr>

<tr>

<td>

<label for="itemtype">Item Category</label>

</td>

<td>

<select id="itemtype" name="itemtype">

<option id="category">Electronics </option>

<option id="category">Attires</option>

<option id="category">Toys</option>

<option id="category">Non Poisonous Food</option>

</select>

</td>

</tr>

<tr>

<td>

<label for="location">Service Type</label>

</td>

<td>

<input type="radio" class="radio" name="servicetype" id ="type1" required="required" value="Normal"> Normal

<input type="radio" class="radio" name="servicetype" id="type2" required="required" value="Express"> Express

<input type="radio" class="radio" name="servicetype" id="type3" required="required" value="Platinum"> Platinum

</td>

</tr>

<tr>

<td>

<input type="submit" value="Book My Consignment" name="submit" id="submitButton" onclick="costcalculation()">

</td>

<td>

<input type="reset" value="Clear" name ="clear" id="clear">

</td>

</tr>

</table>

</form>

<div name="result" id="result"></div>

<table id="costtable">

<thead>

<tr>

<th>Service Type</th>

<th>Weight (in Kg)</th>

<th>Amount(per kg)</th>

</tr>

</thead>

<tbody>

<tr><td rowspan="3">

Normal

</td>

<td>5 to below 15</td>

<td>125</td>

</tr>

</td>

<tr>

<td>15 to below 25</td>

<td>175</td>

</tr>

<tr>

<td>Above 25 to below 36</td>

<td>200</td>

</tr>

<tr><td rowspan="3">

Express

</td>

<td>5 to below 15</td>

<td>175</td>

</tr>

</td>

<tr>

<td>15 to below 25</td>

<td>225</td>

</tr>

<tr>

<td>Above 25 to below 36</td>

<td>250</td>

</tr>

<tr><td rowspan="3">

Platinum

</td>

<td>5 to below 15</td>

<td>225</td>

</tr>

</td>

<tr>

<td>15 to below 25</td>

<td>375</td>

</tr>

<tr>

<td>Above 25 to below 36</td>

<td>400</td>

</tr>

</tbody>

</table>

<script>

function costcalculation()

{

var type = document.getElementsByName("servicetype");

var weight = document.getElementById("kilogramval").value;

var loc = document.getElementById("location").value;

var tax,amount;

var cost;

for(var i =0;i<type.length;i++)

{

if(type[i].checked)

{

type = type[i].id;

break;

}

}

if(loc == "London - 1800")

{

tax =.1;

}

if(loc == "New York - 3100")

{

tax = .15;

}

if(loc == "Tokyo - 2800")

{

tax= .2;

}if(loc =="Singapore - 1400" || loc =="Mexico City - 3300")

{

tax =0;

}

switch(type)

{

case "type1":

if(weight >= 5 && weight < 15 )

{

amount = 125;

}

if(weight >= 15 && weight <= 25)

{

amount= 175;

}

if(weight >25 && weight < 36)

{

amount = 200;

}

break;

case "type2":

if(weight >= 5 && weight < 15 )

{

amount = 175;

}

if(weight >=15 && weight <= 25)

{

amount= 225;

}

if(weight >25&& weight < 36)

{

amount = 250;

}

break;

case "type3" :

if(weight >= 5 && weight < 15 )

{

amount = 225;

}

if(weight >=15 && weight <= 25)

{

amount= 375;

}

if(weight >25&& weight < 36)

{

amount = 400;

}

break;

}

cost = amount\*weight;

cost = cost + tax\*cost;

cost = Math.round(cost);

document.getElementById("result").innerHTML="Your shipment has been accepted. Amount to be paid is Rs."+cost;

}

</script>

</body>

</html>

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Changes Made | | | |
| V1.0.0 | Initial baseline created on <dd-Mon-yy> by <Name of Author> | | | |
| Vx.y.z | <Please refer to the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed> | | | |
| **Section No.** | **Changed By** | **Effective Date** | **Changes Effected** |
|  |  |  |  |