Introduction

We have chosen the SCENARIO 2 for our assessment. This scenario is about a company, Adenium Eco Park, promotes a special discount rate for all visitors in conjunction with the end-of-year school holiday. Moreover, the discounts were given in two types which is Family and Friends, and School Trips.

The fare tickets details were given as follows.

1. Family and friends:

Visitors	Age	Charges (per person)	Special Discount
Toddler	2-3	RM 20.00	5%
Child	4-17	RM 40.00	10%
Adult	18-60	RM 60.00	15%
Baby, Senior		FREE	

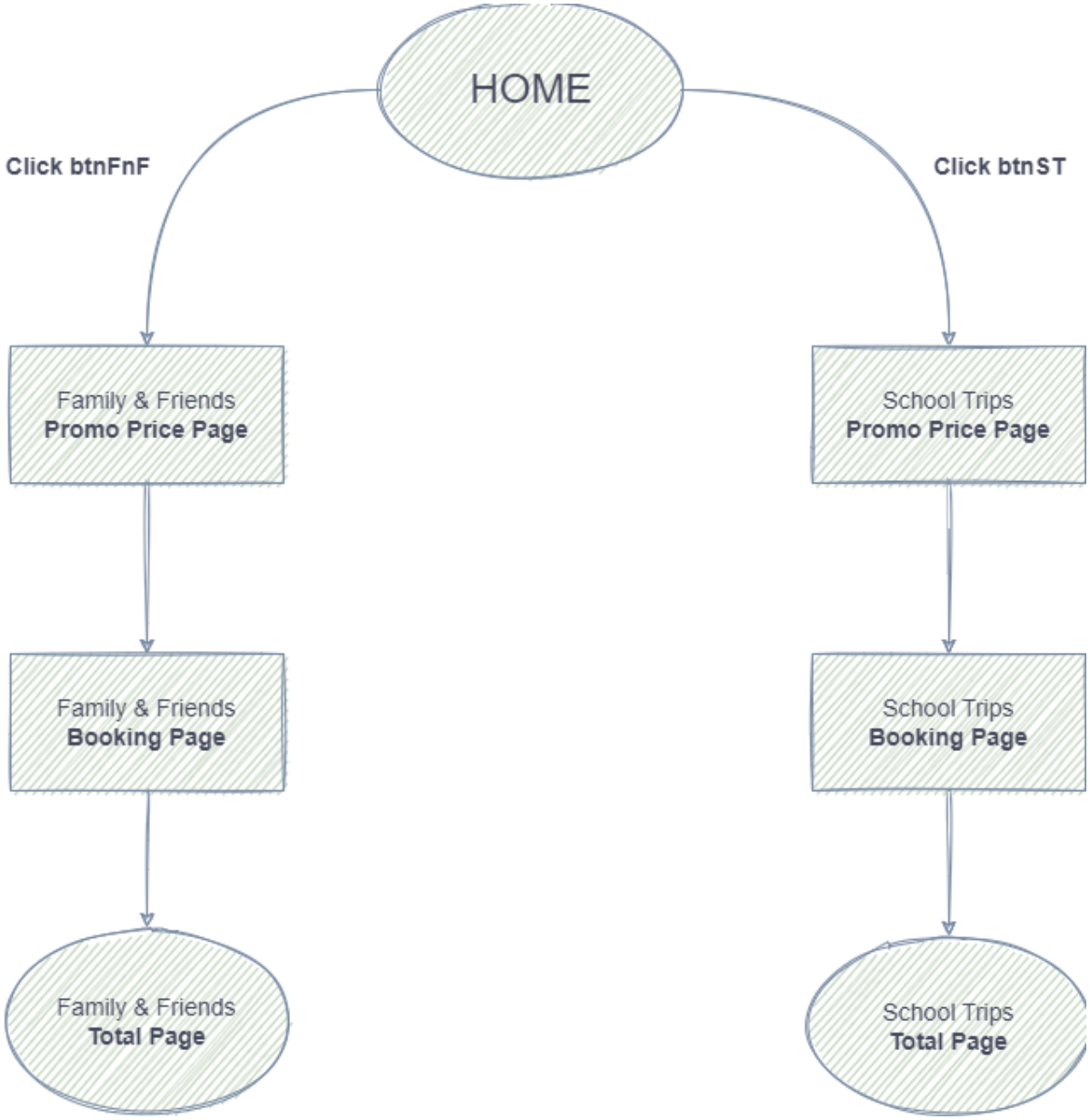
2. School trips:

Visitors	Age	Charges (per person)	Special Discount
Child	4-17	RM 35.00 (minimum 10 students)	15%
Teacher		FREE	

Apart from that, this assessment has also briefed us to *create* a web application that enables a user **to fill in their information** (e.g., name, contact number), **to key in the** fare ticket details, and **to view the details** of the booked ticket, the details price, and the total price.

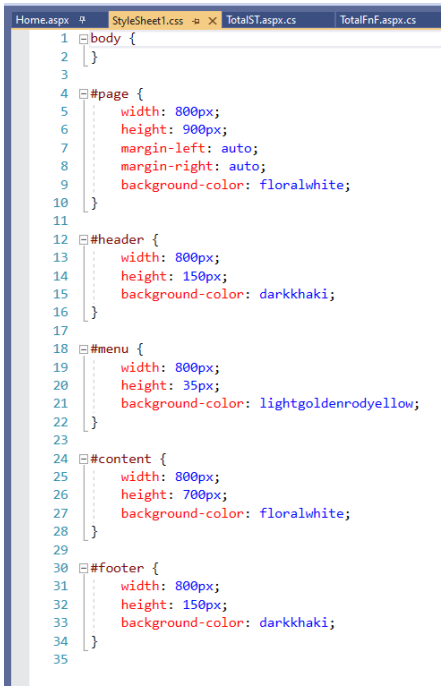
In our web application, we have decided to break it into two parts in order to isolate the Family and Friends(**FnF**) with the School Trips(**ST**) discount rates. Resulting on that, we have come out with 7 pages with 1 home page, 3 FnF pages, and 3 ST pages*, master page excluded*.

Diagram below shows the flow to understand better about our web.



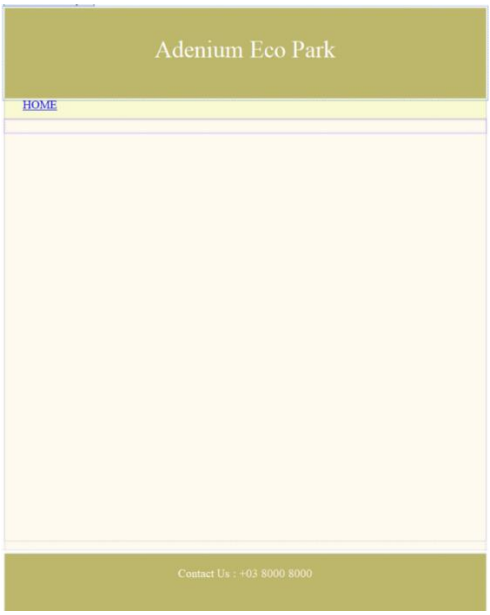
Master Page

In the Master Page, we have customized the stylesheet for a page, header, menu, content and footer with a few soft colours to make the web more eye-friendly to the user/client.



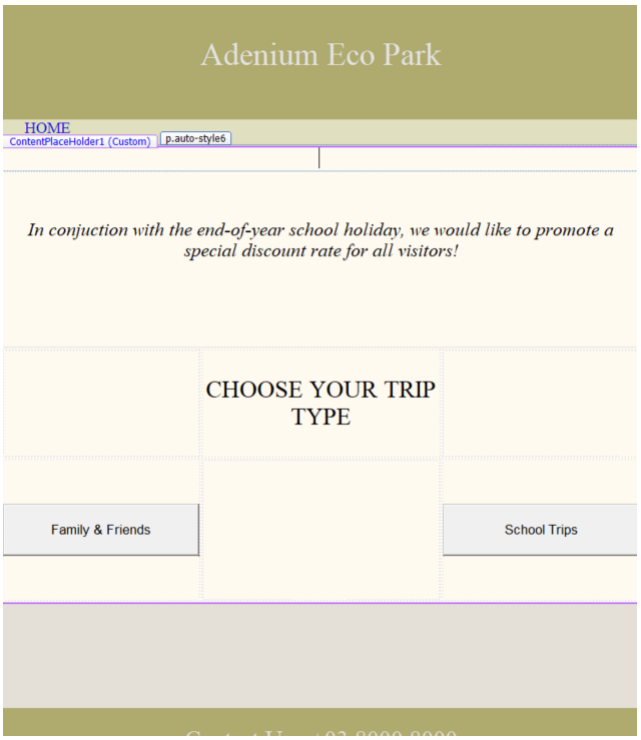

```css
body {
}

#page {
    width: 800px;
    height: 900px;
    margin-left: auto;
    margin-right: auto;
    background-color: floralwhite;
}

#header {
    width: 800px;
    height: 150px;
    background-color: darkkhaki;
}

#menu {
    width: 800px;
    height: 35px;
    background-color: lightgoldenrodyellow;
}

#content {
    width: 800px;
    height: 700px;
    background-color: floralwhite;
}

#footer {
    width: 800px;
    height: 150px;
    background-color: darkkhaki;
}
```

Other than stylesheet, we have also typed in the company name ‘Adenium Eco Park’ in the header, provided hyperlink ‘HOME’ in menu to navigate the page back to home page, and a text ‘contact us’ in the footer to add details.



Home Page

In this page, we have put out the text at the top, ideally to catch user attention to advertise that the company is now promoting a special discount for all visitors. In the middle of the page, we put a text to lead the user into choosing either one of two of the discount rates provided.



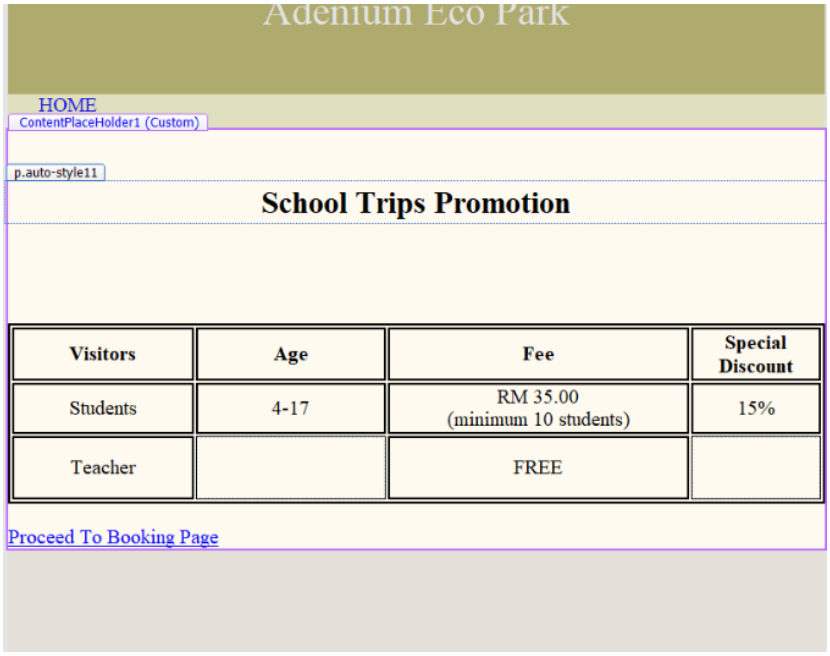
There are two buttons below and each button will direct the user to the into each page. As example, clicking on button School Trips will bring user to the next page for School Trips which is the ST promotion price page

```
0 references
protected void btnFnf_Click(object sender, EventArgs e)
{
    Response.Redirect("promoFnF.aspx");

}

0 references
protected void btnST_Click(object sender, EventArgs e)
{
    Response.Redirect("promoST.aspx");

}
```

Promotion Price Page

In this page, we have basically created a price table to acknowledge users the details of the different discount rates promoted on different age groups.

< Family And Friends



School Trips >

We have also provided a hyperlink below so that user can proceed to key in details and book ticket and the link will navigate them to the booking form page.

Booking Page

For booking form page, it will be different for both of trip type. The **calculation** was also made behind these pages, immediately executed once the user clicked on BOOK NOW.

i) FnF Booking Form

For this type, user will be able to key in their Name, Email, and Phone Number for admin to contact, and details for how many visitors they will bring. We chose to create this kind of design so it will be easy for user to understand and to use it. In addition, we did put some validator on each of those textboxes. There are some with double stars as we put both required field validator and other validator because we considered that they were also the primary key for a form.

a) Required Field Validator:
   All of them

b) Regular Expression Validator:
   - **Email** (Internet e-mail address)
   - **Phone Number** (\d{9})

Below is the source code for the validation made on FnF Booking Page

```
sp:Content>
p:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <table style="width:100%;">
        <tr>
            <td class="auto-style16">Name</td>
            <td class="auto-style16">
                <asp:TextBox ID="txtName" runat="server" Width="320px"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName" ErrorMessage="Please Enter Your Name"
                    ForeColor="Red">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td class="auto-style15">Email</td>
            <td class="auto-style15">
                <asp:TextBox ID="txtEmail" runat="server" Width="245px"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvEmail" runat="server" ControlToValidate="txtEmail" ErrorMessage="Please Enter Your Email"
                    ForeColor="Red">*</asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator ID="revEmail" runat="server" ControlToValidate="txtEmail" Display="None" ErrorMessage="Please Enter
                    ForeColor="Red" ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">*</asp:RegularExpressionValidator>
            </td>
        </tr>
        <tr>
            <td class="auto-style17">Phone Number</td>
            <td class="auto-style17">
                <asp:TextBox ID="txtPhone" runat="server" Width="245px" OnTextChanged="TextBox5_TextChanged"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvPhone" runat="server" ControlToValidate="txtPhone" ErrorMessage="Please Enter Your Phone Number"
                    ForeColor="Red">*</asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator ID="revPhone" runat="server" ControlToValidate="txtPhone" ErrorMessage="Please Enter A Valid Phone
                    ForeColor="Red" ValidationExpression="\d{9}">*</asp:RegularExpressionValidator>
            </td>
        </tr>
```

In the calculation, we had already applied all discount rates onto each visitor. Apart from that, we used Application methods to passes the input from booking Form Page to Total Page. Codes below is the calculation in our program.

```
protected void btnSubmit1_Click(object sender, EventArgs e)
{
    Application["Name"] = txtName.Text;
    Application["Email"] = txtEmail.Text;
    Application["Phone"] = txtPhone.Text;
    Application["Toddler"] = txtToddler.Text;
    Application["Child"] = txtChild.Text;
    Application["Adult"] = txtAdult.Text;
    Application["Baby"] = txtBaby.Text;
    Application["Senior"] = txtSenior.Text;

    double pToddler = Convert.ToDouble(txtToddler.Text) * 20 * 0.95;
    Application["PriceToddler"] = pToddler;

    double pChild = Convert.ToDouble(txtChild.Text) * 40 * 0.90;
    Application["PriceChild"] = pChild;

    double pAdult = Convert.ToDouble(txtAdult.Text) * 60 * 0.85;
    Application["PriceAdult"] = pAdult;

    Application["PriceBaby"] = 0;
    Application["PriceSenior"] = 0;

    Application["TotalPrice"] = pToddler + pChild + pAdult;

    Response.Redirect("TotalFnF.aspx");
}
```

ii) ST Booking Form

Same as FnF Form, user will be able to key in their Name, Email, Phone Number, and details for how many visitors on this page too. We have also put some validator on each of those textboxes. The difference is that the form requires student to be at least 10 persons.

a) Required Field Validator:
   All of them

b) Regular Expression Validator:
   - **Email** (Internet e-mail address)
   - **Phone Number** (\d{9})

c) Range Validator:
   - **Student** (minimum = 10, maximum = 500)

```
47      <asp:RegularExpressionValidator ID="revEmail" runat="server" ControlToValidate="txtEmail" ErrorMessage="Please Enter A Valid Email"
48          ForeColor="Red" ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">*</asp:RegularExpressionValidator>
49    </td>
50  :r>
51  ">
52    <td class="auto-style6">Phone Number</td>
53    <td class="auto-style6">
54      <asp:TextBox ID="txtPhone" runat="server" Width="245px"></asp:TextBox>
55      <asp:RequiredFieldValidator ID="rfvPhone" runat="server" ControlToValidate="txtPhone" ErrorMessage="Please Enter Your Phone Number"
56          ForeColor="Red">*</asp:RequiredFieldValidator>
57      <asp:RegularExpressionValidator ID="revPhone" runat="server" ControlToValidate="txtPhone" ErrorMessage="Please Enter A Valid Number"
58          ForeColor="Red" ValidationExpression="\d{9}">*</asp:RegularExpressionValidator>
59    </td>
60  :r>
61  ">
62    <td class="auto-style11">Visitor:</td>
63    <td class="auto-style3"></td>
64  :r>
65  ">
66    <td class="auto-style7">Student</td>
67    <td class="auto-style7">
68      <asp:TextBox ID="txtStudent" runat="server" Width="28px"></asp:TextBox>
69      <asp:RequiredFieldValidator ID="rfvStudent" runat="server" ControlToValidate="txtStudent" ErrorMessage="Required"
70          ForeColor="Red">*</asp:RequiredFieldValidator>
71      <asp:RangeValidator ID="rvStudent" runat="server" ControlToValidate="txtStudent" ErrorMessage="minimum 10 students"
72          ForeColor="Red" MaximumValue="500" MinimumValue="10" Type="Integer">*</asp:RangeValidator>
73    </td>
```

^ source code for the validation made on ST Booking Page

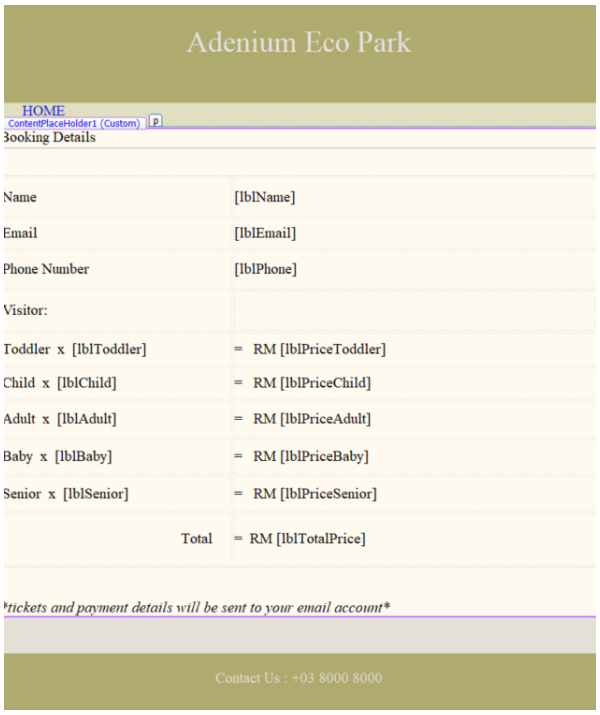
Calculation is similar as before we had already applied all discount rates onto each visitor. Application methods has been used in order to passes the input from booking Form Page to Total Page. Codes below is the calculation in our program.

```
Application["Name"] = txtName.Text;
Application["Email"] = txtEmail.Text;
Application["Phone"] = txtPhone.Text;
Application["Student"] = txtStudent.Text;
Application["Teacher"] = txtTeacher.Text;

double pStudent = Convert.ToDouble(txtStudent.Text) * 35 * 0.85;
Application["PriceStudent"] = pStudent;
Application["PriceTeacher"] = 0;

Application["TotalPrice"] = pStudent;

Response.Redirect("TotalST.aspx");
```

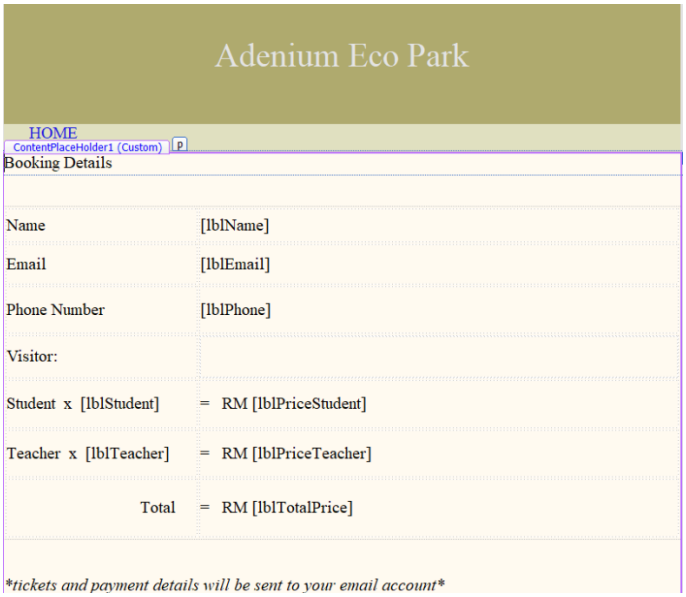
As you can see on some parameter in both calculation, there is equal to zero. It is due to a FREE promotion given on some visitors

Total / Details Page

This page will only appear once the input given by user has passed all validation. On this page, it will briefly show the user information, the details of ticket booked, and the total of all tickets. We have assigned bunch labels to be replaced with the textbox input passed from previous page. Number of labels would be different for each FnF and ST total page.



< Family And Friends



School Trips >

Thank you