|  |  |
| --- | --- |
| **Soal Proyek**  *Project Case* |  |
| COMP6114  Pattern Software Design |
| **Teknik Informatika**  *Computer Science* | **E172-COMP6114-ER01** |
| **Periode Berlaku** Semester Genap 2016/2017  ***Valid on*** *Even Semester Year 2016/2017* | **Revisi 00**  *Revision 00* |

1. Seluruh kelompok tidak diperkenankan untuk :

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from other groups*

* + - Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + - Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | **-** |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Visual Studio 2015 Community Edition  SAP Crystal Reports 13.0.16  IE/Firefox/Chrome |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| Folder Project (SQL, ASPX, ASPX.VB, ASPX.CS, CONFIG, ASHX, WEBINFO, DLL, PDB, DBML,DBML.LAYOUT, DESIGNER.VB, MDF, LDF, image | Folder Project (SQL, ASPX, ASPX.VB, ASPX.CS, CONFIG, ASHX, WEBINFO, DLL, PDB, DBML,DBML.LAYOUT, DESIGNER.VB, MDF, LDF, image |

## Soal

*Case*

**Bluejack Cake**

**Bluejack Cake** is a well-known cake shop in the town. Nowadays, **Bluejack Cake** wants to expand its business by using website. To start this business expansion, it needs to create a website for their shop. In that website, people can find information about the cakes and buy them. This website also allows administrator to manage users, cakes, promotions, and all transactions made by users.

As a web developer, you are asked to create a website using **ASP.net** for **Bluejack Cake**. You are also **required** to use **Domain Driven Design** method to develop the website. The **required** layers are **Controller**, **Handler**, **Repository**, **Factory**, and **Model**. Using web service is **optional** in this project. There are three types of user’s roles in this website: **Administrator**, **Customer**, and **Guest** (**non-logged-in user**).

Minimum pages you need to create for each role are:

* **Administrator**
  + Home
  + Change Password
  + Profile
  + Cake
  + Add Cake
  + Update Cake
  + View Customer
  + Transaction
  + Transaction Report
  + Promotion
  + Add Promotion
  + Update Promotion
  + Logout
* **Customer**
  + Home
  + Change Password
  + Profile
  + Cake
  + Cart
  + Transaction
  + Transaction Report
  + Logout
* **Guest**
  + Home
  + Login
  + Register
  + Forget Password
  + Cake

Each page shows **greeting to user** and **current date time**. The descriptions for each page are:

* + - 1. **Home**

This page tells about general information about **Bluejack Cake** and the **newest** **promotion**. This page is accessible by **all users**.

1. **Login**

This page is **accessible** to **guest** only. Display an **error message** on the login form if user enters wrong combination of **Email** and **Password**. The following table shows the existing fields in the login form and validations for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Email** | * + - Must be filled and match the data in the database |
| **Password** | * + - Must be filled and match the data in the database |

If user has entered correct **Email** and **Password**, the page will be redirected to **Home Page** and if user checks the “**Remember Me**” checkbox, the website will save the **Email** using cookies for **1 hour** and will display the **Email** on the **Email field** in the login form.

1. **Register**

This page allows guests to **register** themselves as **Bluejack Cake** customer. Validate this page is accessible to **guest** only. Display an **error message** if user input incorrect personal data. The following table shows the existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | * + - Must be filled |
| **Email** | * + - Must be filled     - Cannot take an existing Email in database |
| **Password** | * + - Must be filled |
| **Confirm Password** | * + - Must be **exactly same** **as** P**assword** |
| **Date of Birth** | * + - Must be filled |
| **Phone Number** | * + - Must be filled |
| **Address** | * + - Must be filled |

1. **Forget Password**

This page allows user to reset their forgotten **password**. This page is accessible by **guest** user only. Display **error message** if users input **email** that does **not exists** in database. If the **email exists**, a **new password** will be **generated** (the **old password** in the database will be **replaced** with new **password**) and **displayed**. The following table shows the conditions for the **new password**:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **New Password** | Must contain exactly 6 characters  Random each character, as follows:   * Generate two random numbers [0-9] for the first two characters. * Generate two random alphabets [a-z] for the next two characters * The last two characters is obtained from the customer’s name   Example:  **Customer Name: Bella**  **New password: 27zkBe** |

1. **Promotion**

This page will show all promotions created by **administrator** and this page is **accessible** by administrator only. The information of promotion that must be shown are **Promotion Title**, **Description**, and **Discount**.

For administrator user, **Add Promotion**, **Update Promotion,** and **Delete Promotion** buttons will be visible:

* + To **add new promotion**, administrator can click **Add Promotion** button.
  + To **update** **promotion**, administrator can click **Update Promotion** button.
  + To **delete** **promotion**, administrator can click **Delete** **Promotion** button.

1. **Add promotion**

This page allows **administrator** to **insert** **new promotion**. This page will be shown if administrator clicks **Add Promotion** button in **Promotion** page. Display error message if administrator input incorrect promotion data. The following table shows existing fields in this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Title** | * + - Must be filled |
| **Description** | * + - Maximal length is 20 characters |
| **Discount** | * + - Must be filled |

1. **Update Promotion**

This page allows **administrator** to **update the promotion**. This page will be di shown if administrator clicks **Update Promotion** button in **Promotion** page. Display **error message** if administrator input incorrect cake data. The following table shows existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Title** | * + - Must be filled |
| **Description** | * + - Maximal length is 20 characters |
| **Discount** | * + - Must be numeric |

1. **Cake**

This page allows user to see all **cakes information** from **Bluejack Cake**. This page is acccessible for **all users**. The information of cake that must be shown are **Cake Name**, **Cake Picture**, **Stock**,and **Price**.

For **customer** user, this page is also used to buy the cake. Provide fields to input **Chocolate Plate Greeting** (the greeting attached in the cake), **Promotion** (the available promotions),and an **“Add to Cart”** button. Display **error message** if user **doesn’t choose the promotion**, **doesn’t input Chocolate Plate Greeting** or if the **Chocolate Plate Greeting** **consists of** **more than 10 characters**. If the user clicks on “**Add to Cart**” button, the data will be put into the **Cart**.

For **administrator**, provide **Add Cake**, **Update Cake**, and **Delete Cake** buttons:

* + To add new cake, **administrator** can click **Add Cake** button.
  + To **update cake**, administrator can click **Update** button.
  + To **delete** **cake**, administrator can click **Delete** button.

1. **Add cake**

This page allows **administrator** to **insert new cake**. The page will be shown if administrator click **Add Cake** button in **Cake** page. Display **error message** if **administrator** input incorrect cake data. The following table shows existing fields in this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Cake Name** | * + - Must be filled     - Cake Name must be unique (cannot take existing Cake Name) |
| **Price** | * + - Must be filled |
| **Stock** | * + - Must be filled and **greater** than **0** |
| **Picture** | * + - Image **extension** must be **.jpg** or **.png** |

1. **Update Cake**

This page allows **administrator** to **update the cake data**. This page will be di shown if administrator clicks **Update button** in the **Cake page**. Display **error message** if **administrator** input incorrect cake data. The following table shows existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Cake Name** | * + - Must be filled     - Cake Name must be unique (cannot take existing Cake Name) |
| **Price** | * + - Must be filled |
| **Stock** | * + - Must be filled and **greater** than **0** |
| **Picture** | * + - Image **extension** must be **.jpg or .png** |

1. **Profile**

This page allows **customer** and **administrator** to view their personal data. The personal data that must be shown are **Name**, **Email**, **DOB** (Date of Birth), **Phone**,and **Address**.

1. **Change Password**

This page allows **customer** and **administrator** to **change** their **password**. Display **error message** if user input **incorrect** data. The following table shows existing fields in this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Old Password** | * + - Must match with current password in database |
| **New Password** | * + - Must be filled |
| **Confirm Password** | * + - Must be **exactly same** with New Password |

1. **View Customer**

This page is accesible **only** for **administrator**. This page allows administrator to see **all registered Customer** in **Bluejack Cake**. The infomation of customer that must be shown are **Name**, **Email**, **DOB**, **Phone**,and **Address**. Provide **Delete** button to **delete selected customer**.

1. **Cart**

This page is accessible for **customer** only. It allows customer to see the cakes that have been added in the cart. Display cart information such as **Cake Name**, **Cake Price**, **Chocolate Plate Greeting**, **Promotion** (obtained from Promotion’s title), **Discount**, **Sub Total** (Cake Price – Discount), and **Grand Total** (sum of all Sub Totals). Customer **can remove** the cake that have been ordered.

Provide a **Check Out** button. If customer **clicks** Check Out button, the shopping **cart data** will be **inserted to database**, **update** the available **stock** of the cake, and all **cart data** will be **deleted**.

1. **Transaction**

This page is accessible only for **administrator** and **customer**. This page displays all transaction that has been made by customer. The information of transactions that must be shown are **Transaction Date**, **Customer Name**, **Cake Name**, **Cake Price**, **Chocolate Plate Greeting**, **Promotion** (obtained from Promotion’s title), **Discount**,and **Subtotal** (Cake Price - Discount).

For **administrator**, this page **displays** **all** **transactions** **from** **all** **customers**.

For **customer**, this page displays their own **transactions**.

For **administrator** and **member**, provide a **Generate Transaction Report** button. If **administrator** or **customer** clicks the **Generate Transaction Report** button, **administrator** or **customer** will be redirected to **Transaction Report** page.

.

1. **Transaction Report**

This page is accessible for **administrator** and **customer**. This page will be shown if **administrator** or **customer** click the **Generate Transaction Report** link from the transaction page.

For **administrator**, this page **displays** **all transaction from all customers.**

For **member**, this page **displays** **their own** **transactions**.

Here are the rules that you must follow to create your project:

1. Use appropriate software for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
2. Collect appropriate files for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
3. Include the other files that can support your project, such as:
   * All files in your project
   * Other files (image, audio, video, etc.) used in your project
   * \*.DOC file (documentation of your project) that contains the reference links of additional files (image, audio, video, etc.) used in your project

**If you do not understand, please ask your assistant!**