**GAME ADDA**

**No Hackers**

Shubham Shrivastav

Hongfeng Wang

**MIS 5050 Semester Project**

**Milestone 4: Final Project Submission**

**Project Report:**

This project is an organization independent concept that works on the principle of E-Commerce.

This project is independent of an organization and serves to provide a platform for customers who prefer purchasing video games online. It is basically a business to customer model through which gamers will have an open choice of wide variety of video games of different genre and different gaming platform (PC, PS3, PS4, X BOX 360) according to their choice. A customer can make a choice of his own, make an online payment to purchase the game.

User of the website can make tweets on the website and can also can contact directly to the owner of the site by sending mail through the website.

**Application Guide:**

**Master Page:**

1. MasterPage.master: This is primary master page for the site. This page includes navigation bar for navigation through the website, it also contains sidebar so that the user can view latest tweets and upcoming games. All the aspx pages are inherited by this master page, except PayPal page.

**.aspx Pages**

1. Home.aspx: this page is first page of the website from where the user can navigate using the navigation bar.
2. Category.aspx: This page contains a list view control which have all the products which a user can see. The games are displayed on the basis of the platform they selected from navigation bar.
3. GameDetail.aspx: As the Games are listed on Category.aspx, when a user selects a game he will be redirect to the GameDetail.aspx page which contains a Form view and it contains complete information about the selected game. It contains a button so that the user can add the game to his shopping cart.
4. Featured.aspx: This has list view which give the list of featured games on the home page.
5. CartPage.aspx: when user clicks the add to cart button in the GameDetail.aspx page he will be send to CartPage.aspx page, there he can see his selected item in a Grid view, displaying the quantity and price of the product.
6. Login.aspx: To buy a product a user must have authentic account on the website, if he have an account he can proceed, otherwise he will be asked to register on the website.
7. CreateUser.aspx: It is a dynamic page when user click to register he is redirected to this page, which ask the information like username, password , email address about the user and saves it into the websites database so that a user can login using his username and password.
8. ViewProfile.aspx: A user who has account on the website can view his personal information and can update them.
9. UserProfile.aspx: A registered user can view his own profile and can make updates if he wish to.
10. Adimn.aspx: This page is accessible only by the Admin of the website in this page he can view all the games and can update, delete or insert new game. The games are shown using a Grid view and it also contains a detail view to display the selected game and make changes.
11. Tweets.aspx: This page contains a Grid view which displays the comments made by the user regarding games. A user can also post comment by typing in the text box. All the comments are stored on the database.
12. RecoverPassword.aspx: this page allows User to recover his password in case he forgot his password using his username.
13. ContactUs.aspx: This dynamic page allow a user to contact the admin, if user wants to ask the admin anything about the game he can contact the admin by sending him a mail through this page.

**User Controls:**

1. CurrentUserinfo.ascx: A user control that displays the user profile and the username. It also have functionality to update the user’s profile.

**Class Files:**

Cart.cs: This class is used to create the shopping cart for the user which contains the list of the products that the user selects to buy. It is used by the CartPage.

Prod.cs: This class give the public property of the selected product that displays on the CartPage.aspx.

**Database:**

Database**:** This database contains various tables of the website like Games, tweets, Orders etc.

ASPNETDB:This database manages all membership, role and profile information for the application.

**Role Description:**

1. Administrators

Page Access: Admin.aspx, ViewProfile.aspx.

Sample Account:

User name: admin

Password: admin1!

2. User

Page Access: ViewProfile.aspx.

Sample Account:

User login ID: user

Password: user123!

**Project Requirements:**

|  |  |
| --- | --- |
| Project Requirements | Project Components |
| 5 interlinked Web forms (.aspx pages) that display dynamically-generated content per team member. | Home.aspx, Category.aspx, GameDetail.aspx, CartPage.aspx, Featured.aspx, Admin.aspx, ContactUs.aspx, CreateUser.aspx, Login.aspx, ViewProfile.aspx, RecoverPassword.aspx, LatestTweets.aspx |
| At least 5 database transactions (a transaction includes selecting, inserting, updating, or deleting data from the database) per team member | 1. Category.aspx: Select all games from Games table of selected platform. 2. GameDetail.aspx: Selects a Game, Which user wants to see in detail. 3. Featuted.aspx: selects the featured games by fetching data from database. 4. CartPage.aspx: It displays the list of games that user want to buy. 5. LatestTweets.aspx: Insert command and select command is used to insert in Tweets table. 6. CreateUser.aspx: Insert command is used to insert details of user in database. 7. ContactUs.aspx: Insert command to provide feedback and ask questions. 8. Admin.aspx: This page uses select, update, insert, and delete command. 9. ViewProfile.aspx: Update and insert command is used. Here, a user can view his profile and can update it. 10. Recoverpassword.aspx: Here user can recover the password. |
| User/Role management for at least 2  roles | 1. Administration: Admin can login as administrator and can upload new game or can update. 2. Customer: If a user logins he can access the website see games can buy them, and can view his own profile. |
| Preservation of state (may include  cookies, session state, application  state, cross-page posting, or  querystring) | For preservation of state we have used sessions. |
| 10 different ‘basic’ ASP.NET Web  controls (those that map directly to an  html tag) | We used TextBox, Label, image control, DropDown List, HiddenField, Button, FileUpload, HyperLink, Wizard, CalenderExtender. |
| 5 different ‘advanced’ ASP.NET Web  controls (those that do not map  directly to an html tag) | Three different Login controls, GridView, DetailView, FormView. |
| input validation for all free-form user  entries | The pages in which user have to fill data have validators so that if data is not fill or if data is invalid error message is displayed. |
| 1 user control | CurrentUserInfo.ascx is used. |