**Team: *Voltron***

**Members: *Kevin Zhao & Jordan Bills***

**Project: *Glenn’s Electric Ecommerce Website***

**Application Summary:** Our application is an ecommerce website specifically designed for Glenn’s Electric, which is a local store here in Logan. There will be to main users of this website. The first one will be the customer and the second the owner or manager of the store. For the customer the application provides pages for them to view different products by category and purchase the product right from the application. It will also allow them to create a profile so they can save their information each time they buy something from the store. On the owner side they will be able to see everything the customer see as well as additional pages to manage categories and products. They will also be able to respond to customer inquiries.

**Application Guide:**

1. Class Files
   1. Cart.cs - class that contains code to interact with the cart.
   2. Cartitem.cs - class that contains code to interact with items inside the cart.
2. Databases
   1. Glenns.mdf - database that includes data about products, categories, and order information.
   2. IdentityDB.mdf - database that includes data about users that create an account as well as the different roles.
3. Other aspx pages
   1. CategoriesPage.aspx - page that dynamically shows images and names of all categories in the project.
   2. Checkout.aspx - page that shows fields for user to enter in address and payment information.
   3. ContactUs.aspx - page that shows fields for user to fill out to send the admin a message.
   4. Default.aspx - landing page for users that has information about Glenn’s Electric.
   5. ProductDetails.aspx - page that shows image and product details for whatever product the user clicks on.
   6. Login.aspx - page that allows users to login into their profile.
   7. ProductPage.aspx - page that shows images and name of all products filtered by the category the user clicks on.
   8. ViewCart.aspx - page that shows the items and total price of items in the cart.
   9. Profile.aspx - page that allows users to update their profile information.
   10. Register.aspx - page that allows users to create a new profile.
   11. CategoryManager.aspx - page that allows admin to update, delete, and create categories for the products in the store.
   12. ProductManager.aspx - page that allows admin to update, delete, and create products in the store.
   13. Response.aspx - page that allows admin to respond to customers who fill out and submit the fields on the contact page.
   14. ErrorHandling.aspx - page that shows simple message and contact information for administrator when the website runs into an error.
   15. Confirmation.aspx- page that displays order that was placed
4. Master Pages
   1. MasterPage2.master - this is the master page for all pages in the project. It contains login and cart header as well as navigation buttons and a footer.
5. Other Supporting Files
   1. Web.config - xml file that contains information about website such as connection strings and what pages each role or user can access.

**Role Description:**

1. User
   1. Page Access: Default.aspx, CategoriesPage.aspx, ProductPage.aspx, ProductDetails.aspx, Checkout.aspx, ViewCart.aspx, Login.aspx, Profile.aspx, Register.aspx
   2. Sample User Account
      1. UserName: test123
      2. Password: test123
2. Admin
   1. Page Access: CategoryManager.aspx, ProductManager.aspx, Reponse.aspx, plus all pages User has access to.
   2. Sample User Account
      1. UserName: adminuser
      2. Password: admin123

|  |  |
| --- | --- |
| 5 interlinked Web forms (.aspx pages) that display dynamically-generated content per team member (see project description document for details). | Jordan   * CategoryManager.aspx * ProductManager.aspx * CategoriesPage.aspx * ProductsPage.aspx * ProductDetails.aspx   Kevin   * ContactUs.aspx * Response.aspx * Checkout.aspx * ViewCart.aspx * Confirmation.aspx |
| At least 5 database transactions (a transaction includes selecting, inserting, updating, or deleting data from the database) per team member | Jordan   1. CategoriesPage.aspx    1. Select statement to display categories 2. ProductsPage.aspx    1. Select statement with where clause from query string to display products by category 3. ProductDetails.aspx    1. Select statement to query details about product user selects 4. CategoryManager.aspx    1. Select statement to display all categories    2. Delete category from database    3. Insert category into database   Kevin   1. ContactUs.aspx    1. Insert into form into database 2. Response.aspx    1. Inserts Response into database.    2. Select statement to show messages from customers 3. Checkout.aspx    1. Inserts Form into Orders    2. Inserts Form into Order Details |
| User/Role management for at least 2 roles | Roles Supported   1. Admin 2. User |
| Preservation of state (may include cookies, session state, application state, cross-page posting, or querystring) | When Admin or Customer logs in, their userid is stored in the session. So other user pages such as response.aspx or customerhistory.aspx can be accessed. |
| 10 different ‘basic’ ASP.NET Web controls (those that map directly to an html tag) | Fnamelabel (label)  FirstName (textbox)  Submitbutton (button)  Loggingin (hyperlink)  Billing\_country (dropdown)  Ship\_checkbox (checkbox)  Image1 (image)  ContentPlaceHolder1 (PlaceHolder)  hiddenPrice (HiddenField)  Validation1 (Validation) |
| 5 different ‘advanced’ ASP.NET Web controls (those that do not map directly to an html tag) | LoginView1 (login)  ListView1 (listview)  GridView1 (Gridview)  DetailsView1 (Detailsview)  SqlDataSource2 (SqlDataSource) |
| Input validation for all free-form user entries | ContactUs.aspx   * Make sure that all fields are entered before you can submit. |