|  |  |  |  |
| --- | --- | --- | --- |
| Unit Title | SD: Developing Websites for Multiplatform Use | | |
| Candidate’s Name | Kieran Burns | Class | HND Soft Dev A |
| **SOFTWARE ENGINEERING DOCUMENTATION: Module Description** | | | |
|  | | | |
| **Function Name** | **Description** | | |
| Index() | This function returns an action result the welcome page review to the User, it takes no parameters. It loads the View html code to the front end where the user can interact with the page. | | |
| Login() | This function also returns an action result and takes in two parameters, users username and password which the user enters. After this the credentials are checked against the database, to see if the users input matches. If the details are correct the user is logged in an redirected to the welcome page. If the details are not valid then an error message will pop up, asking the user to enter their details again. | | |
| RegisterUser() | This function allows a new unregistered user to create an account on the web application. This function also returns an action result & takes 1 parameter: an object model, that will hold the user’s name, email, address, post code, password and password confirmed which will all be collected from the user when they enter their details in the front end. If entered correctly, then the information is added to the database and the user is redirected to the home page. If the registration is not successful then the user is prompted to fix errors where they appear to be. This function will have a validation check if the password is secure and if the user already exists in the database. | | |
| CreateAPost() | This function will allow the logged in member/staff user to post a blog by taking in one parameter that passes an object which will have the users blog information: Title, category, content. Once the information has been entered if successful the post will be posted, if not successful an error message will popup describing what went wrong or If it crashes, then the user will be met by a http page not found message. | | |
| EditPost() | This function will allow the registered user who is logged in to edit their own post. It will take in the post ID as a parameter to identify the post and display it, but when the page is being changed we use polymorphism to overload this method (and use HTTPPost annotation) using an object as a parameter and the object passed in will be the users, blog title, content, comments and category, these are the elements the user may potentially be changing, any changes made will be saved and stored in the database. Any errors, will either return a pop up, http bad request or page not found. | | |
| Details() | This function will allow registered user to view their own posts details, passing in one parameter, the posts id, which we use to find the selected post in the table. If the post cant be found then we return either a http bad request or page not found. | | |
| DeletePost() | This function will allow the user to delete the selected post and any post comments, it will also display the details for the post being deleted. It takes in one parameter, which is a id, to find the post by ID then the category of the post. If there is an error loading the page, then the user will get either a http bad request error or page not found. | | |
| DeleteConfirmed() | This function is an HTTPPost and will trigger after the user hits the ‘delete’ button. This function also takes an ID as the parameter and searches for the post by id then removes and saves the changes made to the database. After all is said and done the user will be redirected to the ‘My Blog Posts’ page. | | |
| Comment() | This function will allow only registered users to leave a comment on a post. It will take in a ID as a parameter, to find the ID of the user trying to make a comment. The comment will be added, and the comment will be saved in the database. Any errors will display a pop up or a page not found. | | |
| SuspendAUser() | This function will allow only Admins to suspend users from commenting. It will take in a ID as a parameter to find the ID of the user that the admin is trying to suspend. After user is suspended they can still log in, but their ability to leave a comment will be switched to false. Again if any errors occur, a display popup or page not found will be displayed. | | |
| PromoteAUser() | This function will allow Admins to promote a user by assigning the user a different role. It will take in a ID as a parameter to find the ID of the user that the admin is trying to promote. The user will be assigned all the authorisations/permissions of their new role. If any errors occur, a display popup or page not found will be displayed. | | |
| PersonalisePosts() | This function allows the Admin to add a picture to a staff member's post. Which will be posted just above the “Posted 2 months ago…” text to personalise a staff members post. It will take in a ID as a parameter to find the ID of the staff user that the admin is trying to personalise. It will allow the admin to upload an image, which the staff can request for a change in person if they want. If any errors occur, a display popup or page not found will be displayed. | | |