EXPERIMENT – 4

TESTING

AGRICULTURAL MANAGEMENT SYSTEM

AIM:

In this experiment we discuss in brief about different types of testing, and provide mechanisms to have hands-on experience on unit testing.

SOFTWARE TESTING:

Testing software is an important part of the development life cycle of a software. It is an expensive activity. Hence, appropriate testing methods are necessary for ensuring the reliability of a program. According to the ANSI/IEEE 1059 standard, the definition of testing is the process of analyzing a software item, to detect the differences between existing and required conditions i.e. defects/errors/bugs and to evaluate the features of the software item.

The purpose of testing is to verify and validate a software and to find the defects present in a software. The purpose of finding those problems is to get them fixed.

Verification is the checking or we can say the testing of software for consistency and conformance by evaluating the

results against pre-specified requirements.

Validation looks at the systems correctness, i.e. the process of checking that what has been specified is what the user actually wanted.

Defect is a variance between the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development (coding) phases.

PROJECT DESCRIPTION:

Product Perspective

Hardware Interface

Hard disk: The database connectivityrequires a hardware configuration that is on-line. This makes it necessary to have a fast database system running on high rpm hard disk permitting complete data redundancy and back-up systems to support the primary goal of reliability. The system must interface with the standard output device, keyboard and mouse to interact with this software.

Software Interface

Front End : HTML5, CSS, PHP

Back End : PHP,Apache, WAMP server With MYSQL

Memory Constraints

No specific constraints on memory.

Operations

The software allows two modes of operations

- Enquire about the availability of goods and services
- By extracting the username and password the software allows the user to sell the products.

Product Functions

Enquire about the availability of products. Search the availability of product by entering the product name. The software validates the authentic user by extracting their user name and password. After the validation of the user software allows the farmers to sell their products and wholesalers to buy the products.

User Characteristics

The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user .The Product is absolutely user friendly, so the intended users can be the naïve users. The product does not expect the user to possess any technical background. Any person who knows to use the mouse and the keyboard can successfully use this product.

Constraints

The user can login only using their unique username and password. In case , the user has forgotten the password ,it can be retrieved through their registered email.

Types of Software Testing

- **UNIT TESTING**
- **INTEGRATION TESTING**
- **SYSTEM TESTING**

1)Unit Testing:

Unit testing is done at the lowest level. It tests the basic unit of software, that is the smallest testable piece of software. The individual component or unit of a program are tested in unit testing. Unit testing are of two types.

1. Black Box Testing
2. White Box Testing

TEST SUITE:

Module 1: Verify "User Login" functionality

#	TS1
Title	Verify "User Login" functionality
Description	To test the different scenarios that might arise while an user is trying to login

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC1	Verify that user already registered with the AMS is able to login with correct user ID and password		CUSTOMER ID *1234* is a registered user of AMS; user's password is *this_is_password*	User is logged in	1. Type in customer id as *1234* 2. Type in password *this_is_password* 3. Click on the 'Login' button	"Home" page for the user is displayed
TC2	Verify that an unregistered user of AMS is unable to login		customerID *1566XX* is not a registered user of LIS	User is not logged in	1. Type in customer ID as *1566xx* 2. Type in password *whatever* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*

#	TS1
Title	Verify "User Login" functionality
Description	To test the different scenarios that might arise while an user is trying to login

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC3	Verify that user already registered with the AMS is unable to login with incorrect password		customer ID is a registered user of AMS user's password is *this_is_password*	User is not logged in	1. Type in customer ID as *4567* 2. Type in password *whatever* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*
TC4	Verify that user already registered with the AMS is unable to login with incorrect password given twice consecutively	TC3	This test case is executed after execution of TC3 before executing any other test case	User is not logged in	1. Type in employee ID as *4567* 2. Type in password *whatever2* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*
TC5	Verify that user already registered with the AMS is unable to login with incorrect password given thrice consecutively	TC4	This test case is executed after execution of TC4 before executing any other test case	User is not logged in	1. Type in customer ID as *4567* 2. Type in password *whatever3* 3. Click on the 'Login' button	The "Login" dialog is shown with a *error invalid user credentials*

Module 2: Verify "search according to your needs" functionality

2#	TS2
Title	Verify "search according to your needs" functionality
Description	To test the different scenarios that might arise while an customer/buyer/wholesaler is trying to search for a product

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC1	Select the desired product variety from the select filter/list		Desired product is fruit. If we select fruit it displays the variety of fruits available.	Varieties of fruits displayed.	4. Select the variety from the list. 5. Click on the 'GO' button	"RESULT" page for the variety is displayed
TC2	Search without selecting any desired module from the filter/list.		Displays all the product available.	Varieties of all products are displayed	4. Without selecting any variet. 5. Click on the 'GO' button	"RESULT" page for variety of all products is displayed.
TC3	Search for selected product details.	TC2	This test case is executed after execution of TC2 before executing any other test case	It displays the product image,product sub-variety,price and button to buy,button to add cart and review.	4. Click the product image.	Displayed the result page
TC4	Verify that the user buy the product/add the product in cart/write review	TC3	This test case is executed after execution of TC3 before executing any other test case	If we buy,it displays transaction details. If we cart the product then it is added to cart .If we write a review it is submitted or submitting a review without enter anyting ,review point is taken as 0.	4. Click 'BUY' button 5. Click 'ADD TO CART' button 6. Click on 'submit'	1. If we buy,it displays transaction details 2. If we cart the product then it is added to cart 3. If we write the review it is submitted.

Module 3: Verify " upload product" functionality

#	TS3
Title	Verify " upload product" " functionality
Description	To test the different scenarios that might arise while an customer/buyer/wholesaler is trying to upload a product

#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC 1	user want to upload a product variety from the given category		User should registered in the AMS website.	Product uploaded successfully.	6. Click upload product 7. Click choose file and upload the product 8. Type the product name 9. Type the price of the product 10. Enter submit button	" Digital market page is displayed"
TC2	User upload a product without the product name		User should registered in the AMS website.	User is not uploaded the product	6. Without typing product name 2. Enter submit button	The result page show error you cannot upload product with this extension.
TC 3	User upload a product without choosing the product image file		User should registered in the AMS website	User is not uploaded the product	5. Without uploading the product image 6. Click the submit button	The result page show error you cannot upload product with this extension
TC4	User upload a product without typing the price of the product		User should registered in the AMS website	User is not uploaded the product	7. Without typing the price 2.Click the submit button	. The result page show error you cannot upload product with this extension

Module 4: Verify " BUY PRODUCTS" functionality

#	TS4
Title	Verify " BUY PRODUCTS" " functionality
Description	To test the different scenarios that might arise while an user is trying to buy a product

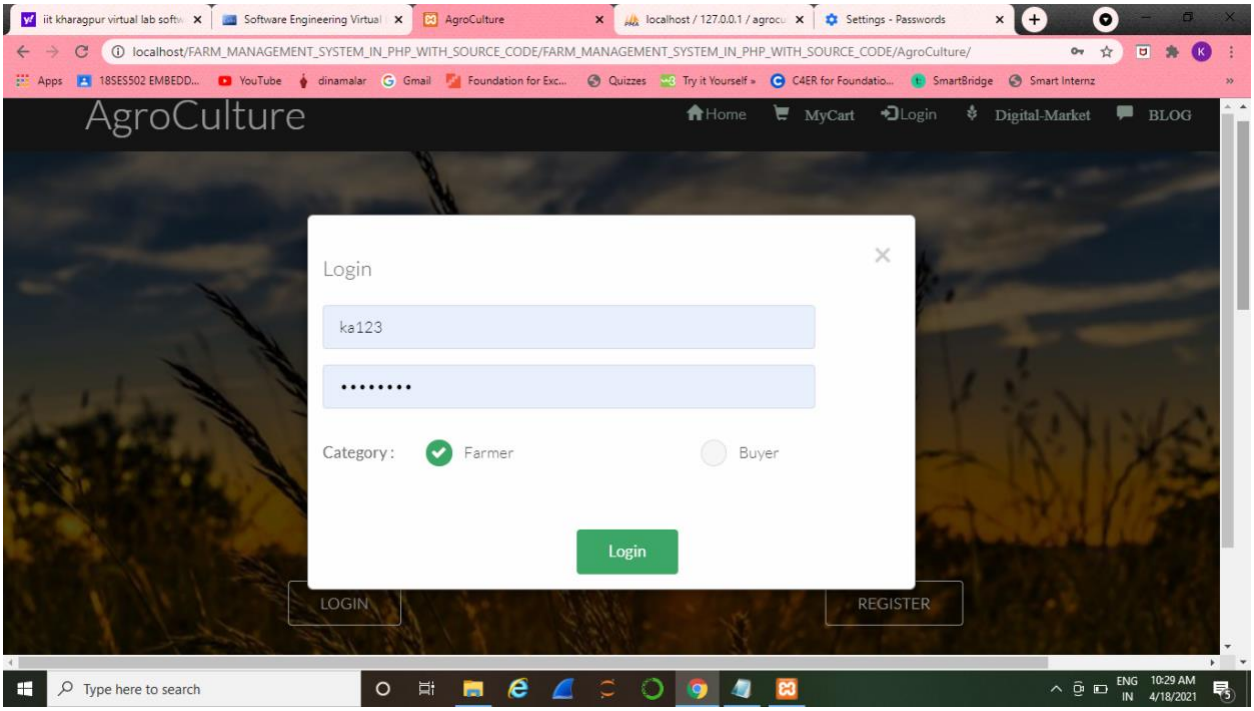
#	Summary	Dependency	Pre-condition	Post-condition	Execution Steps	Expected Output
TC 1	Buyer want to buy a product variety from the given category		buyer should registered in the AMS website.	Order placed successfully	11. Select the product from the given category 12. Click buy now\add to cart 13. 3. Fill the transaction details 14. 4.enter confirm order	" Result page displayed with SUCCESS ..order successfully placed!!
TC2	Buyer wants to buy a product without filling the Necessary fields in the transaction details	TC1	Buyer should registered in the AMS website.	Buyer should not placed the order	7. Without typing necessary field in the transaction details 2. Enter confirm order	The result page show error you cannot placed the order.

BLACK BOX TESTING:

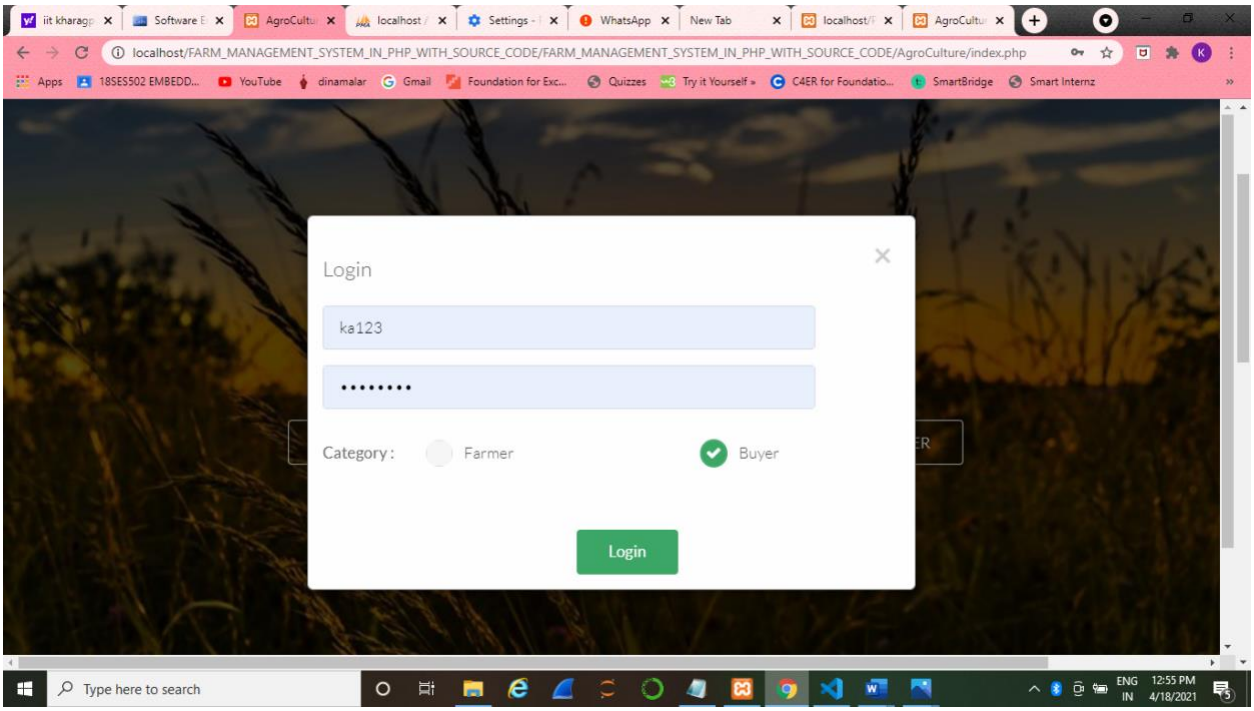
This is also known as functional testing , where the test cases are designed based on input output values only. There are many types of Black Box Testing but following are the prominent ones.

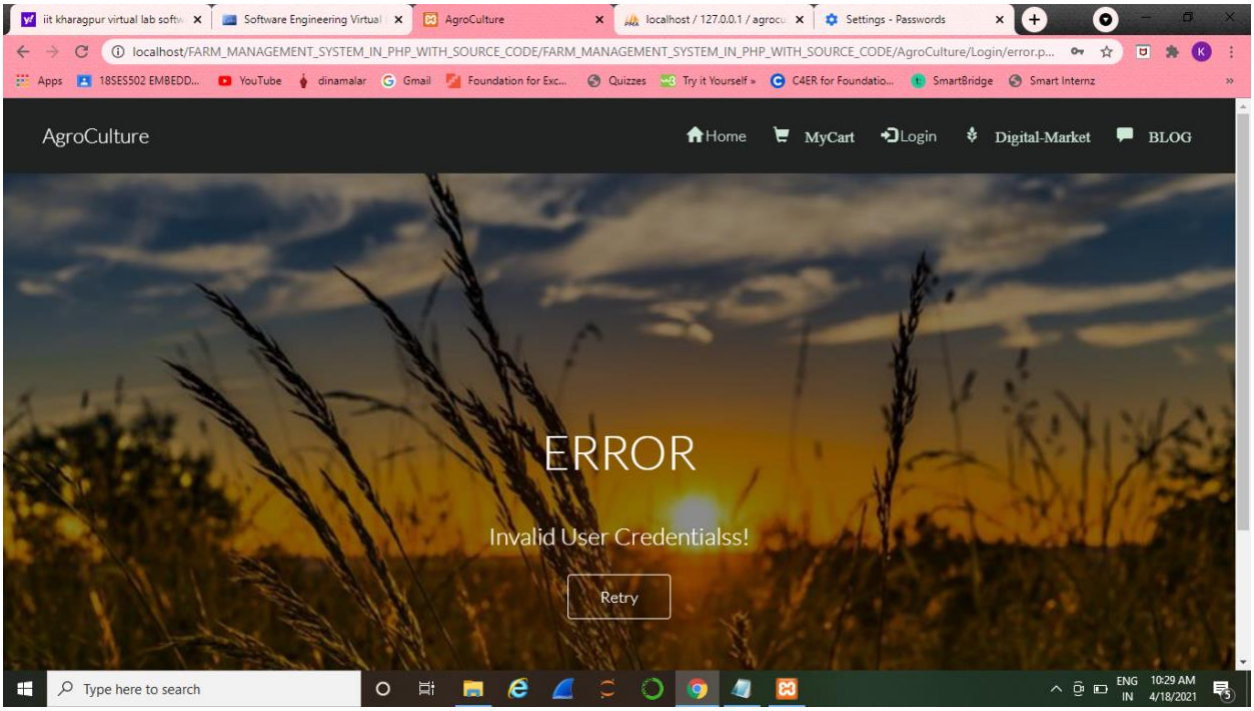
Equivalence class partitioning: In this approach, the domain of input values to a program is divided into a set of equivalence classes. e.g. Consider a software program that computes whether an integer number is even or not that is in the range of 0 to 10. Determine the equivalence class test suite. There are three equivalence classes for this program. - The set of negative integer - The integers in the range 0 to 10 - The integer larger than 10.

Boundary value analysis : In this approach, while designing the test cases, the values at boundaries of different equivalence classes are taken into consideration. e.g. In the above given example as in equivalence class partitioning, a boundary values based test suite is { 0, -1, 10, 11 }

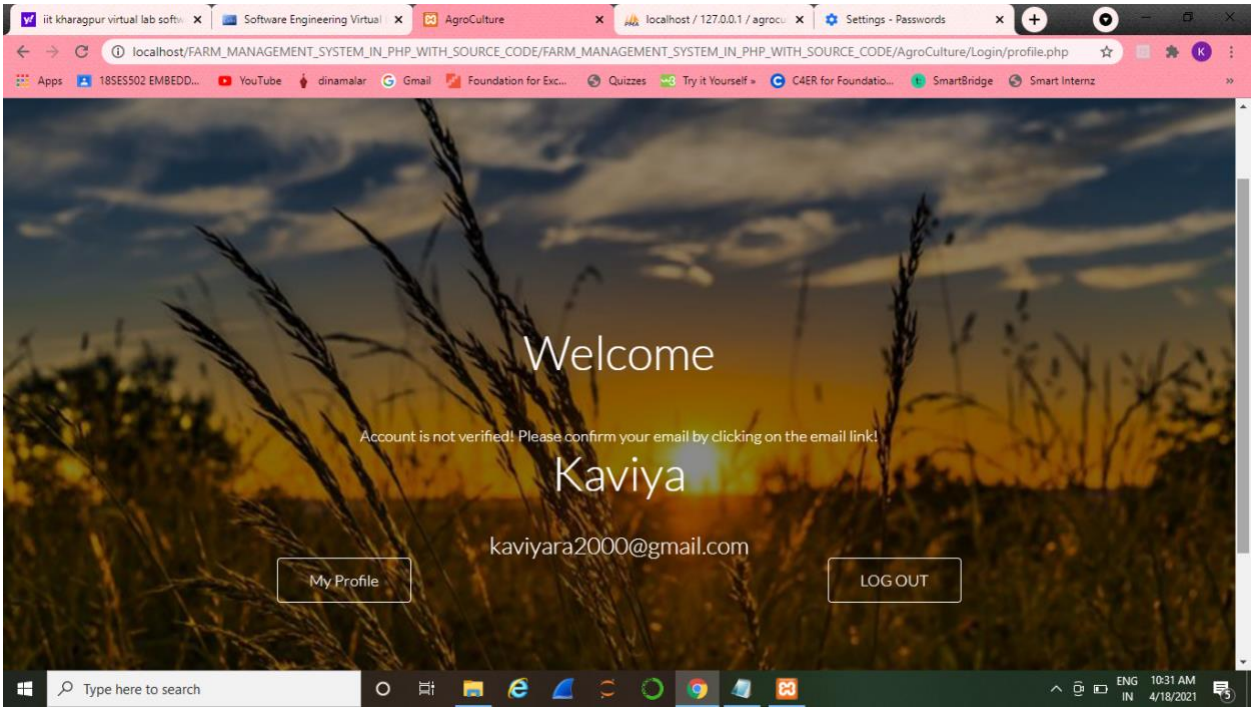


Here this user ka123 is registered as farmer. If we change this domain to buyer to login,it shows invalid user credentials.

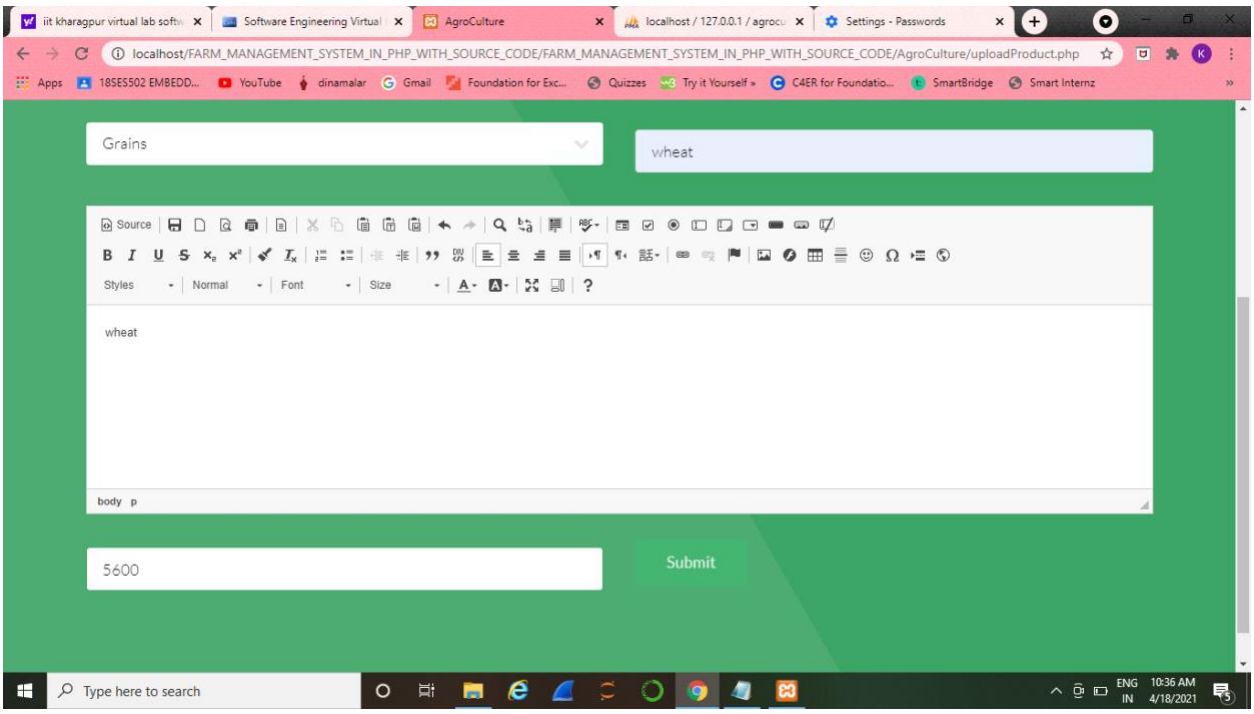


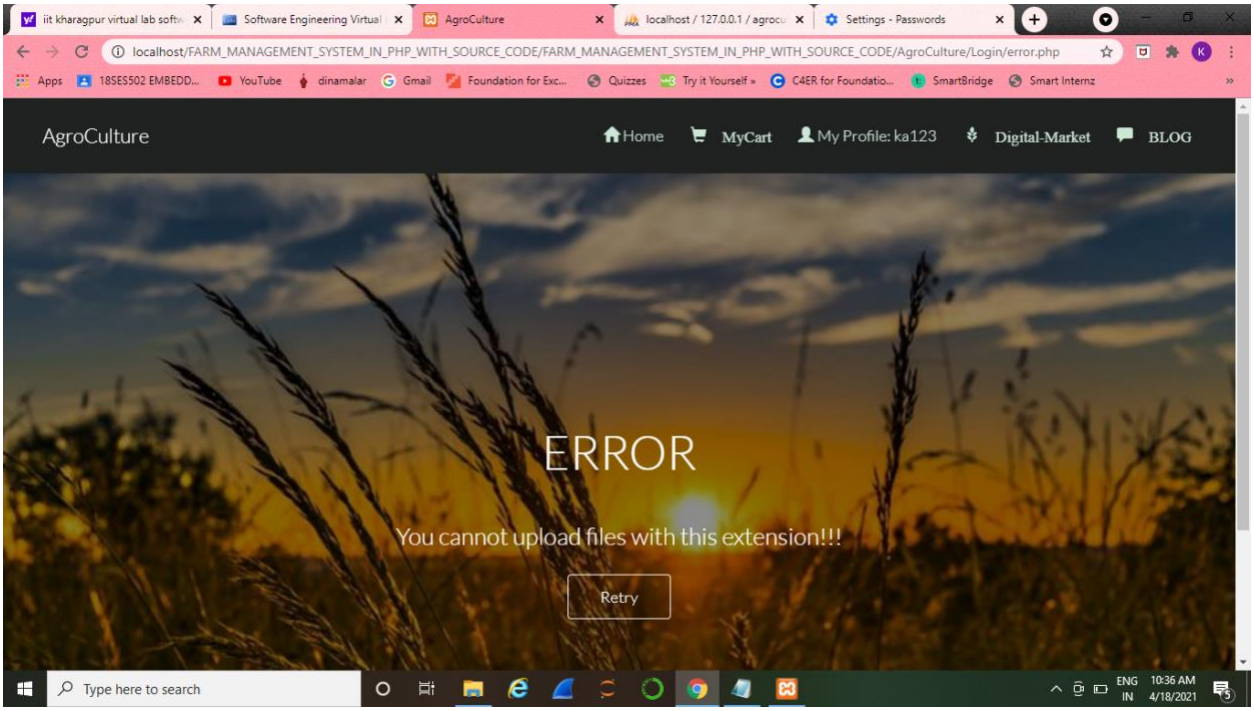


2.While registering,once the registration is completed it shold sent an confirmation mail to registered mail id.But notification has showed and no confirmation mail has received.So from testing this test case is failed.



3.During uploading products with all information ,it shows “you cannot upload file with this extension”.This is an another issue found by black box testing.



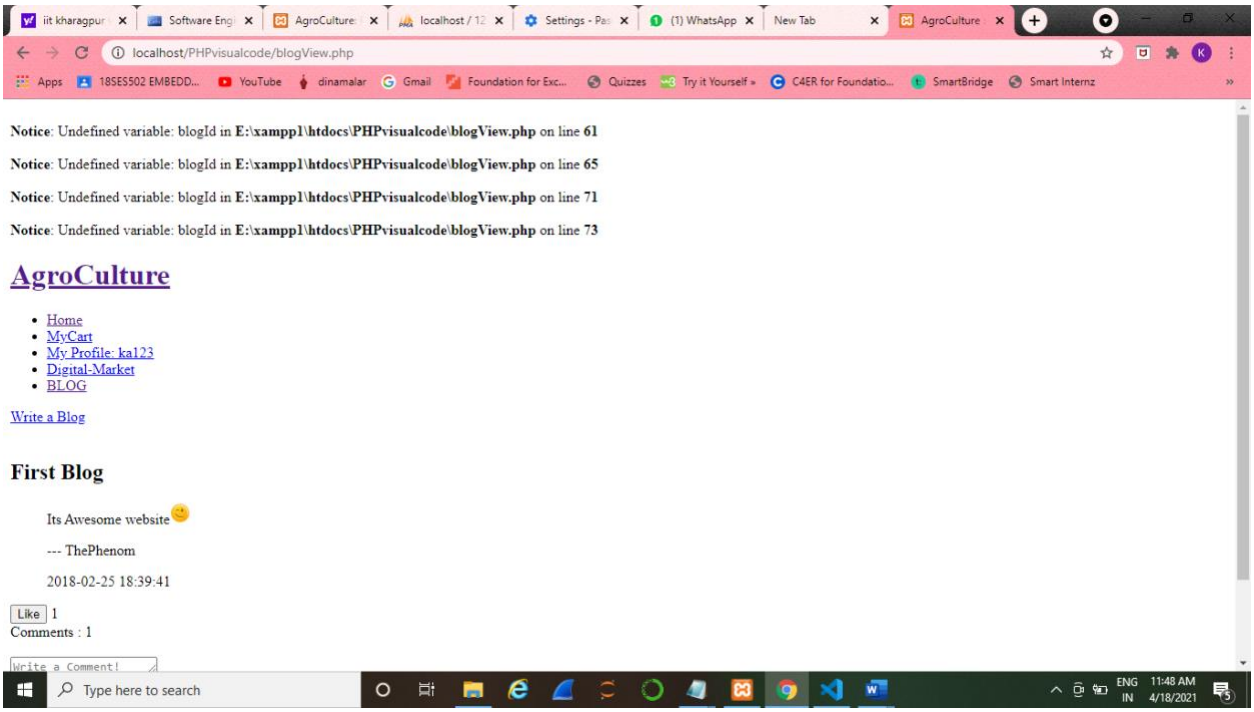


2.WHITE BOX TESTING:

It is also known as structural testing. In this testing, test cases are designed on the basis of examination of the code.This testing is performed based on the knowledge of how the system is implemented. It includes analyzing data flow, control flow, information flow, coding practices, exception and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover

exploitable vulnerabilities.This testing requires access to the source code. Though white box testing can be performed any time in the life cycle after the code is developed, but it is a good practice to perform white box testing during the unit testing phase.

1.Logical error in viewBlog.php .this is found by white box testing



2.Error in view.php is found by white box testing.

Parse error: syntax error, unexpected '}', expecting end of file in **E:\xampp1\htdocs\PHPvisualcode\blogView.php** on line **36**

After removing all the errors

AgroCulture

- Home
- MyCart
- My Profile: ka123
- Digital-Market
- BLOG

Write a Blog

First Blog

Its Awesome website

--- ThePhenom

2018-02-25 18:39:41

Like 1
Comments : 1

Write a Comment!

Submit

AvatarThePhenom
Mast yarr 6:39 pm

2) Integration Testing

Integration testing is performed when two or more tested units are combined into a larger structure. The main objective of this testing is to check whether the different modules of a program interface with each other properly or not. This testing is mainly of two types:

1)Top-down approach

2)Bottom-up approach

In bottom-up approach, each subsystem is tested separately and then the full system is tested. But the top-down integration testing starts with the main routine and one or two subordinate routines in the system. After the top-level 'skeleton' has been tested, the immediately subroutines of the 'skeleton' are combined with it and tested.

3)System Testing

Alpha testing is done by the developers who develop the software. This testing is also done by the client or an outsider with the presence of developer or we can say tester.

Beta testing is done by very few number of end users before the delivery, where the change requests are fixed, if the user gives any feedback or reports any type of defect.

User Acceptance testing is also another level of the system testing process where the system is tested for acceptability. This

test evaluates the system's compliance with the client requirements and assess whether it is acceptable for software delivery

An error correction may introduce new errors. Therefore, after every round of error-fixing, another testing is carried out, i.e. called regression testing. Regression testing does not belong to either unit testing, integration testing, or system testing, instead, it is a separate dimension to these three forms of testing.