



INTELLIGENT DEBUGGING EVALUATOR WITH ADAPTIVE TEST ALGORITHM

A PROJECT REPORT

Submitted by

JEFFIN BEN MATTCAKAL (311011104035)

JEWEL JAMES (311011104036)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KCG COLLEGE OF TECHNOLOGY, KARAPAKKAM

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2015

ANNA UNIVERSITY : CHENNAI 600 025**BONAFIDE CERTIFICATE**

Certified that this project report **“INTELLIGENT DEBUGGING EVALUATOR WITH ADAPTIVE TEST ALGORITHM”** is the bonafide work of **“JEWEL JAMES (311011104036) and JEFFIN BEN MATTACAKAL (311011104035)”** who carried out the project under my supervision.

\SIGNATURE

Mr. Cloudin. S., M.E

HEAD OF THE DEPARTMENT

Associate Professor
Dept. of Computer Science
KCG College of Technology,
Karapakkam-97.

INTERNAL EXAMINER**SIGNATURE**

Mrs. Dhanya Anand., M.E

SUPERVISOR

Assistant Professor
Dept.of Computer Science
KCG College of Technology,
Karapakkam-97.

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First of all we thank the God almighty for giving us the opportunity to complete our project successfully; we thank our beloved parents and friends for their encouragement

We express our hearty thanks to Late **Dr. K. C. G. Verghese**, Founder and **Dr. Elizabeth Verghese**, Chairperson for giving us a chance to get into this reputed institution.

We heartily express our sincere and special thanks to our principal **Dr.T.Rengaraja Ph.D.**, who gave us a full support and encouragement to complete our project.

We also express our hearty thanks to **Mr. S. Cloudin M.E.**, Head of the Department, for his commendable inspiring guidance to do the project.

We would particularly like to thank and express our deepest gratitude to our internal guide **Mrs. Dhanya Anand M.E.**, Assistant Professor , Department of Computer Science and Engineering for her complete dedication, support, advice and encouragement in the completion of this project.

We also express our deep sense of gratitude and profound thanks to all our beloved lecturers for the constant encouragement to the complete this project work successfully.

ABSTRACT

Goal of this project is to create an online evaluator tool that is capable of evaluating the marks secured from the debugging test and display the result in the web browser itself. This system also has higher end features which will increase the effectiveness of the test module and then increase the efficiency of the programmers.

In real time system, these website are capable of only executing a string comparator mechanism. This Mechanism is not effective when the system results of the debugging are considered. These online compliers are only compilers system, not online evaluating system.

This design algorithm segregates the available programs in database to easy, hard and medium based on number of errors, number of characters, number of loops, number of functions, number of variables and number of statements used. This algorithm module helps to improve programming level of learner. The adaptive function will test the user based on their performance and helps the user to improve their programming skills in that particular programming language.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
	LIST OF FIGURES	vii
	LIST OF TABLES	viii
	LIST OF ABBREVIATIONS	ix
1.	INTRODUCTION	
	1.1 OVER VIEW OF THE PROJECT	1
	1.2 ABOUT THE PROJECT	2
2.	LITERATURE SURVEY AND SYSTEM ANALYSIS	
	2.1 LITERATURE REVIEW	3
	2.2. EXISITING SYSTEM AND ITS DISADVANTAGES	6
	2.3. PROPOSED SYSTEM AND ITS ADVANTAGES	7
	2.4. HARDWARE AND SOFTWARE REQUIREMENTS	8
	2.5. FEASIBILITY STUDY	9
	2.6. PROJECT SCHEDULE (GANTT CHART)	9
	2.7. SCOPE OF THE PROJECT	9
3.	SYSTEM DESIGN	
	3.1. ARCHITECTURAL DESIGN	10
	3.1.1. Architectural Diagram	10
	3.1.2 Design Modules	11
	3.2. USER INTERFACE DESIGN	13
	3.2.1. Use Case Diagram	14
	3.3 DATA DESIGN	15
	3.3.1 Design Description	15
	3.3.2 Data Flow Diagram	15

3.4	MODULAR DESIGN	16
3.4.1	Description	16
3.4.2	Activity Diagram	18
3.4.3	Sequence Diagram	
4.	IMPLEMENTATION AND TESTING	
4.1.	LIST OF ALGORITHM	20
4.1.1	Adaptive Test Algorithm	20
4.2.	ALGORITHM DESCRIPTION	21
4.2.1	Flow Chart	21
4.3.	TEST PLAN	22
4.4.	TEST PROCEDURES	22
4.5.	TEST CASES	22
5.	RESULTS AND DISCUSSION	26
6.	CONCLUSION AND FUTURE WORK	27
	APPENDIX - 1: SAMPLE CODE	28
	APPENDIX - 2: SCREENSHOTS	50
	REFERENCES	55

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
2.1	Architecture of debugger	4
2.2	Project Schedule	9
3.1	Architectural diagram of the online debugger	11
3.2	Use Case Diagram	14
3.3	Data flow of the debugging system	16
3.4	Activity Diagram	18
3.5	Sequence Diagram	19
4.1	Flow Chart	21

LIST OF TABLES

+

TABLE NO.	TABLE NAME	PAGE NO.
2.1	Literature Survey Analysis	6
2.2	Comparison of Existing and proposed System	8
4.1	Test Case 1	22
4.2	Test Case 2	24
4.3	Test Case 3	25

LIST OF ABBREVIATIONS

RF	Random Function
ATA	Adaptive Test Algorithm
ODE	Online Debugging Evaluator
OTP	One Time Password
DSC	Display Source Code

CHAPTER 1

INTRODUCTION

The online Debugging Evaluator is a Web Based Project that is a website developed using PHP that can compile C and C++ programs in the web browser. This is also an online evaluator tool that is capable of evaluating the marks secured from the debugging test and displays the result in the web browser itself. It also has an Adaptive Test Algorithm that can organize the preceding debugging questions based on your performance in the previous questions. This algorithm segregates the available programs in the database in to Easy, Hard and Medium based on the no. of errors, no. of characters, no. of functions used, no. of loops used, no. of variables used and no. of statements used. Hence this module helps to improve the programming level of the learner.

1.1 OVERVIEW OF PROJECT

The Online Debugging Evaluator is a Web Site that is Capable to Debug C and C++ Programs and then evaluate them using the Adaptive Test Algorithm and Random Function.

Objective Of the Project

- To build an online debugging evaluator with adaptive test module.
- The adaptive test module algorithm is capable of alternating between the question patterns based on the user performance.
- A random function is used to randomly choose the programs for the test module
- The evaluator module will contain the compiler and the score card.

PROCESS INVOLVED IN THE PROJECT

The Major processes involved in this project are:-

XAMP

1. It helps to create a website with an easier user interface.
2. It helps to use a local host.

INPUT

1. Test programs in a Turbo C, C++

PROCESS

1. Adaptive Function.
2. Online Compiler to compile the programs.
3. Random function to choose the programs.

OUTPUT

1. Score Card is generated after evaluation.

1.2 ABOUT THE PROJECT

The main purposes of the project are mentioned below:

- There are so many symposiums and technical events conducted in various engineering colleges which require an online debugging evaluator
- In software companies also there is a huge opportunity for online debugging system for trainees.
- So to meet the technical needs of so many software firms and engineering colleges, I we are trying to develop this online software tool.

Hence this project module will be a Corporate or university level Training tool for programmers.

CHAPTER 2

LITERATURE SURVEY AND SYSTEM ANALYSIS

2.1 LITERATURE REVIEW

A Web Registry for Publishing and Discovering Mathematical Services

Rebhi Baraka, Olga Caprotti, and Wolfgang Schreiner

Research Institute for Symbolic Computation (RISC-Linz)

Johannes Kepler University, Linz, Austria

PAPER'S CONTENT:

This paper contain the Mathematical web service Registry which has the following modules.

ENTITIES INVOLVED IN THE DEBUGGER: The main entities involved are:

- Problem
- Algorithm
- Implementation
- Realisation
- Machine Type

Problem

It is the algorithm that is used to produce the mathematical algorithms. The mathematical problems that need a solutions are first gathered together and then these mathematical problems are sent to the algorithm one by one to get solution.

Algortihm

It is the main algorithm or the idea that is making the web service work. The algorithm can easily create the solution for all the mathematical problems that are sent to it by using the mathematical laws and functions used.

Implementation

The implementation part has the GUI and the other website interfaces. The implementation part has the local host server that hosts all the server side scripts and the domain side scripts are viewed in the browser.

Realisation

Realisation part identifies the errors in the coding part. All the programs are first compiled and then these programs are filtered with respect to the number of the errors available in any program.

DISPLAYING MODULE

This is a separate website used to display result.

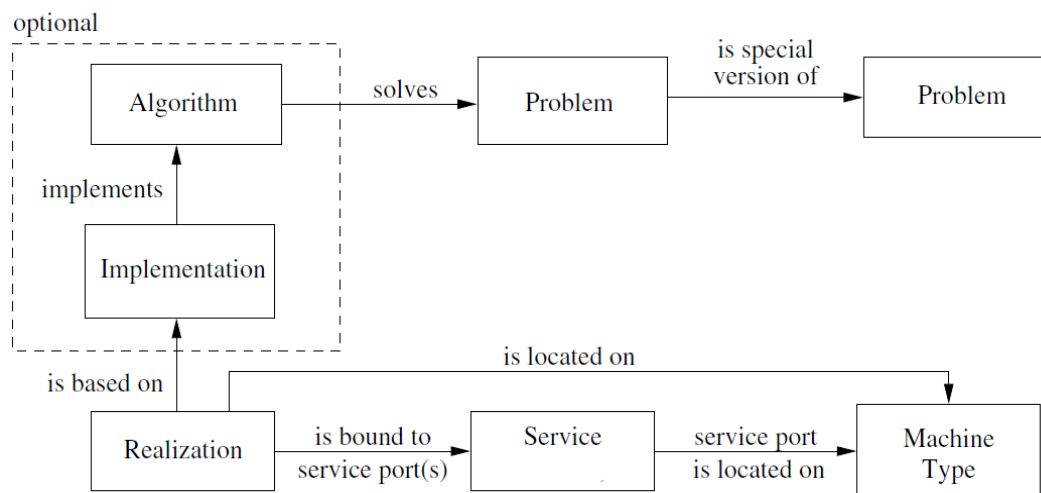


Figure 2.1 Architecture of debugger

MACHINE TYPE:

ENTITIES INVOLVED IN THE EVALUATOR:

The main entities involved are:

- Mark Database
- Score Module
- Evaluator
- Mark arranger
- Display Module

MARK DATABASE

The mark module is used only to calculate the marks of students. The database used is a MySQL database. This database is storing the details of all the students with the marks scored by the particular student.

SCORE MODULE

The score module is used to calculate the test scores. The Score Module evaluates the programs as per the amount of time consumed to debug the program, number of compilations done, and number of errors solved.

EVALUATOR

The evaluator finally calculates the entire score. The Score is calculated using an algorithm that uses the amount of time consumed to debug the program, number of compilations done, and number of errors solved to calculate the score.

MARK ARRANGER

The mark arranger will have the algorithm to arrange the users according to their marks. This algorithm is very quick and very effective. This also uses algorithms to sort the top scorers as per the amount of time consumed to debug the program, number of compilations done, and number of errors solved to calculate the score.

Table 2.1 Literature survey analysis

Paper	Paper Name	Algorithm Used	Advantage	Disadvantage
2007	Configurable Composition and Adaptive Provisioning of Web Services Quan Z. Sheng, Member, IEEE, Boualem Benatallah, Member, IEEE, Zakaria Maamar, and Anne H.H. Ngu	MoRSed	Adaptive provisioning for web services	No runtime discovery
2011	runtime service discovery. Automated Software Engineering, Mahbub, K., Spanoudakis, G. & Zisman, A. (2011).	AI planning system	Runtime discovery	Efficiency is low
2011	Self-healing Cycle for Web Service Composition K.S. May Chan	MAPE loop	Self healing of web service	Less efficiency
2012	A Web Registry for Publishing and Discovering Mathematical Services Rebhi Baraka*, Olga Caprotti, and Wolfgang Schreiner	MathBroker	Web registry using MSDL	No runtime discovery
2012	Proactive and Reactive Runtime Service Discovery: A Framework and Its Evaluation Andrea Zisman, George Spanoudakis, Member, IEEE,	Push and Pull Algorithm	Proactive and reactive runtime discovery	no proper linking between proactive and reactive discovery

2.2 EXISTING SYSTEM AND ITS DISADVANTAGES

The existing Systems that are available for online debugging are also websites. These websites are capable of only executing a string comparator mechanism. This mechanism is not effective when the system results of the debugging are considered. Few best examples of online compiler available are

- IDEONE.COM
- W3SCHOOLS.COM

Drawback's in existing system:

These Online compilers are only a compiler system and they do not have an online evaluating system. And these online compilers lack the following features:

- A dedicated Evaluator
- Lack of actual compiler
- Results are not accurate

2.3 PROPOSED SYSTEM AND ITS ADVANTAGES

The proposed ONLINE DEBUGGING EVALUATOR contains the following modules

- Online Compiler,
- Online Examination Module
- Online Evaluator with Mark sheet.

This system also has higher end features which will increase the effectiveness of the test module and then increase the efficiency of the programmers. These features are:

- Adaptive Test Algorithm
- Random Function

ADDITIONAL FEATURES:

The main features which are not present in the existing system are:

- Online Evaluator
- Online Mark Sheet
- Examination Module
- Adaptive Test Algorithm
- Random Function

Table 2.2 Comparison of existing and proposed system

EXISTING SYSTEM:	It only has an online compiler	It does not have a adaptive test module.	Both modules are not available together.
PROPOSED SYSTEM:	It has an online compiler with a evaluator system with mark sheet.	It has an adaptive test module.	Online Debugging Evaluator is available with adaptive test module.

2.4 HARDWARE AND SOFTWARE REQUIREMENTS

The minimum requirements for this software are

- Pentium 4 1.7Ghz Processor with 256mb RAM.
- Windows XP
- Xamp local host server
- LAN network or WiFi

TECHNOLOGIES USED

PHP

It is used for web service designing. It is a web based language. It is used for browser side scripting.

JAVASCRIPT

It is used to do the validation of the website. It is used as a browser side scripting language.

HTML

All the website front end is developed using the HTML. The graphics and the main GUI is created using HTML.

XAMP

Xamp is used to create a local host and to host the website in it. The xamp is used to do all the execution of the entire project.

2.5 FEASIBILITY STUDY

This project is an Application oriented project that can be implemented using a simple Local Host server and PHP coding with the HTML coding. Hence this project Modules are easy to develop and implement. All the modules of the ONLINE DEBUGGING EVALUATOR System has very less complications and hence this is a feasible project with the involvement of less hardware and simple software tools.

2.6 PROJECT SCHEDULE (GANTT CHART)

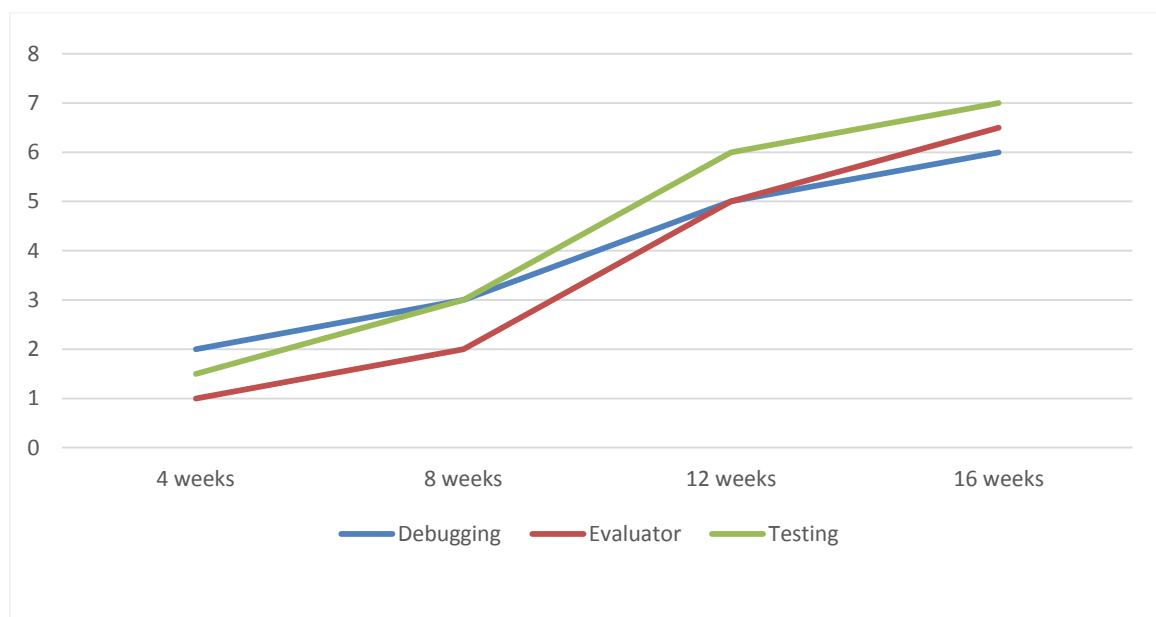


Figure 2.2. Project Schedule

2.7 SCOPE OF THE PROJECT

The Project has a very wide scope. Few examples of the application of this project are:

- There are so many symposiums and technical events conducted in various engineering colleges which require an online debugging evaluator
- In software companies also there is a huge opportunity for online debugging system ,for trainees.
- So to meet the technical needs of so many software firms and engineering colleges, this project helps to develop this online software tool.
- It is also necessary to discover related web services. So this module will be a helpful to for all these requirements.

CHAPTER 3

SYSTEM DESIGN

1. ARCHITECTURAL DESIGN

The architectural design involves the following modules in it. They are:

Entities involved in the debugger:

The main entities involved are:

- Programs
- Adaptive Algorithm
- Implementation
- Random Function
- Evaluator

Entities involved in the evaluator:

The main entities involved are:

- Mark Database
- Score Module
- Evaluator
- Mark arranger
- Display Module

3.2 ARCHITECTURAL DIAGRAM

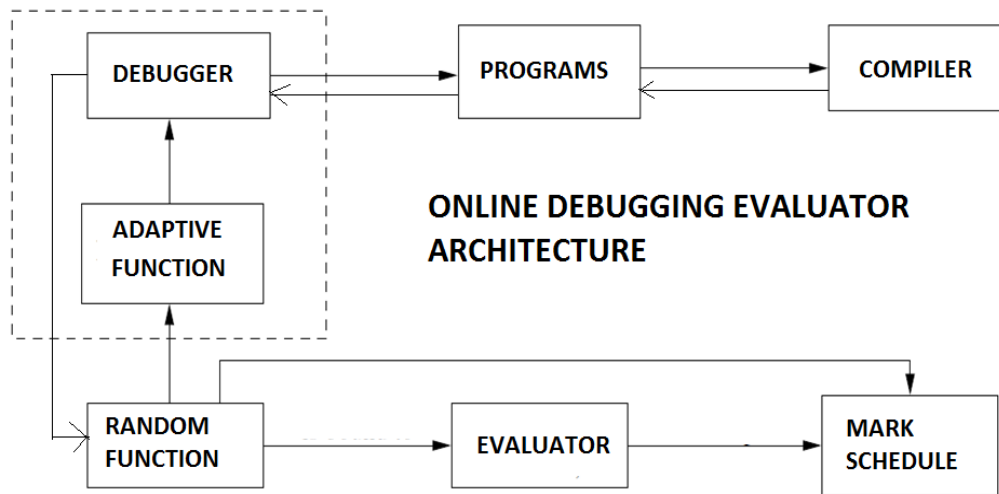


Figure 3.1 Architectural diagram of the online debugger

3.2.1 Design Modules

The major design modules available are:

- Compiler
- Debugger
- Evaluator
- Program
- Adaptive Function
- Random Function

COMPILER

The compiler used in this ONLINE DEBUGGING EVALUATOR is the Turbo C compiler which is called into the browser through the environmental path which is configured in the server.

DEBUGGER

The Debugging Evaluator module is used to compile the programs and detect the number of errors available. This module will finally detect the maximum number of errors found in any program and then send the result to the evaluator module.

EVALUATOR

The Evaluator module is an important module that calculates the marks scored by the participant in the module. This has a separate algorithm to calculate the marks scored by the participant by using the amount of time consumed for debugging, number of errors available and the amount of toughness in the program that is compiled.

PROGRAM

The programs that are needed for compilation are used are kept inside a separate folder and then finally the programs are rearranged based on the number of errors, no. of lines, no. of functions used, no. of loops used, no. of pointers used, no. of arrays used and the no. of loops used.

ADAPIVE FUNCTION

Adaptive function is the core module of this project. This adaptive function is capable of sorting the programs as per the user condition like beginner, intermediate and expert category. This module will categorize the programs as per the user and then changes the programs in run time based on the user performance in the previous tests.

ALGORITHM

Step 1: The user input for Beginner, intermediate and expert is taken in a drop down menu.

Step 2: The conditions for sorting the programs based on the user criteria is taken.

Step 3: Now all the programs available for the test module is compiled once and made ready for adaptive function.

Step 4: The programs are categorized based on their number of lines, no of errors, and the number of functions, loops, variables, pointers, arrays and header files used.

Step 5: Now the Adaptive function will separate the programs based on these criteria and then the programs will be segregated into EASY, MEDIUM and HARD Destinations.

$$\text{Pg Weigt.} = \frac{(\text{no. of Fn} * 3) + (\text{no. of loops} * 2) + (\text{no. of ptr} * 1) + (\text{no. of arr} * 1) + (\text{no. of var} * 0.5)}{5}$$

$$\text{Total weight} = \text{No. of errors} + \text{no. of lines} + \text{pg Weight}$$

Step 6: Now the test module will send one normal level program.

Step 7: Based on the user's performance in the previous programs the adaptive function will decide whether to send easy program or hard program.

RANDOM FUNCTION

The Random Function is used to randomly choose between the programs that are available in all the locations of programs. The programs are actually categorized in to three and separated and saved into three destinations

- Easy Programs
- Medium Programs
- Hard Programs

3.3 USER INTERFACE DESIGN

XAMP

1. It helps to create a website with an easier user interface.
2. It helps to use a local host.

INPUT

1. Test programs in a Turbo C, C++

PROCESS

1. Adaptive Function.
2. Online Compiler to compile the programs.
3. Random function to choose the programs.

OUTPUT

1. Score Card is generated after evaluation

3.3.1 Use case Diagram

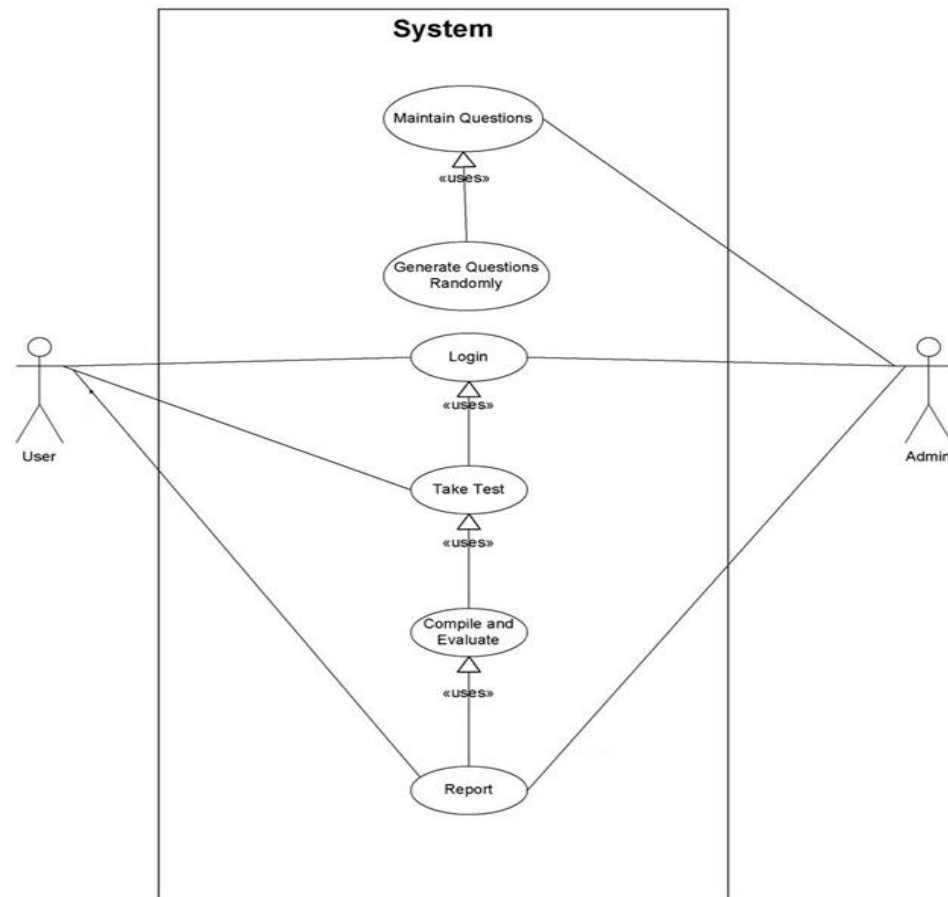


Figure 3.2 Overall flow of the system in Use Case Diagram

The use case diagram describes system functionality as a set of tasks that the system must carry out and actors who interact with the system to complete the tasks. In the use case diagram shown in figure 3.2 all the use cases and the actors are depicted along with the interactions between them. Initially the user gives the password and user name to the login page. Then the user is allowed to take the test. When test gets over, the results are been compiled and evaluated. Then the compiler itself makes the report and sends it to the user. The admin administrates the process and maintain the questions.

3.4 DATA DESIGN

3.4.1 Design Description

DATA FORMATS:

All the programs involved in the modules are in a .txt format. And the module files are all HTML files. The main module execution files are written in php files. The validations are performed using JavaScript files. The types of file formats involved totally are four and they are:

- HTML Files
- Php files
- Txt files
- Javascript files

HTML DATA

The HTML files are used for the front end designing and the GUI interface. These files are saved with .html extension and these files are executed in the browser without any other complicated softwares.

PHP DATA

Php files are all used to do the server side scripting. These files can be executed only with the local host server. These files need an additional software called Xamp to deploy a local host and only then these files can be executed.

TEXT DATA

All the text data that are involved in this project are containing C and C++ programs. These programs are compiled as text files and saved in the same program.

JAVASCRIPT DATA

The javascript files are written to use the validation for all the modules. Hence in this script all the modules will be validated using the javascript files. These files are saved with the .js extension.

3.4.1 Data Flow Diagram

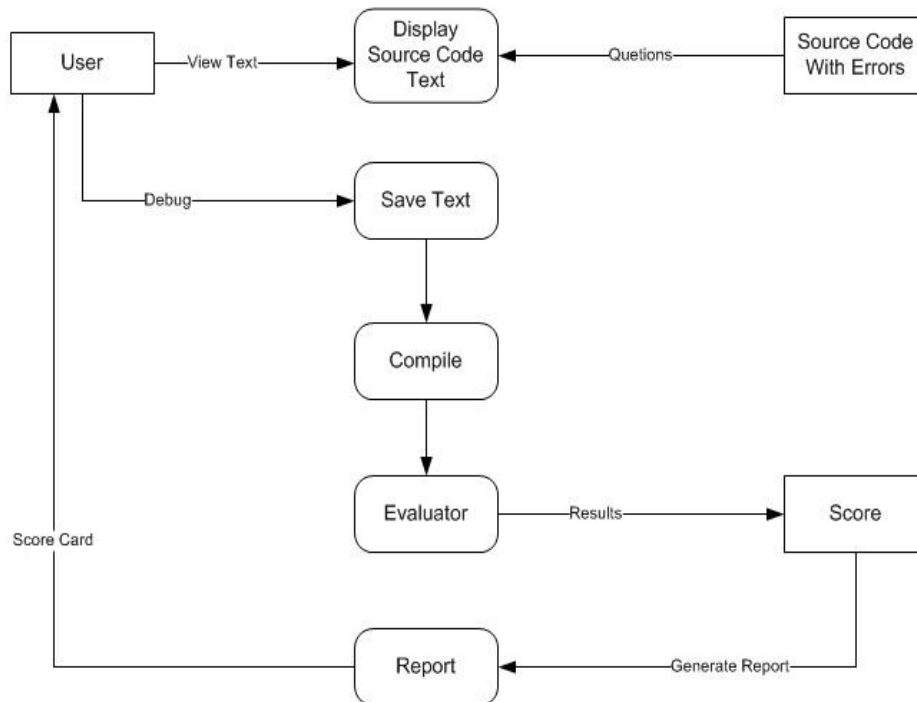


Figure 3.3 Data flow of the debugging system

3.5 MODULAR DESIGN

3.5.1 Description

- Login

This module gets the user name and roll number as input from the user for taking the test. On giving submit, the login details are stored in the database and a new folder is created for that particular user. Simultaneously, the user is navigated to the test page.

- Test

This module displays the source code with errors to the user from the database using the random function initially and adaptive function based on the user's performance in the test.

- Compiler

This module takes the entire source code as input and compiles the code.

- Evaluator

This module takes the number of errors in the source code, time consumed for the test and number of compilations as input and evaluates the result and stores it in the database.

- Report

This module fetches the result from the database and generates the scorecard/report to the user.

3.5.2 Activity Diagram

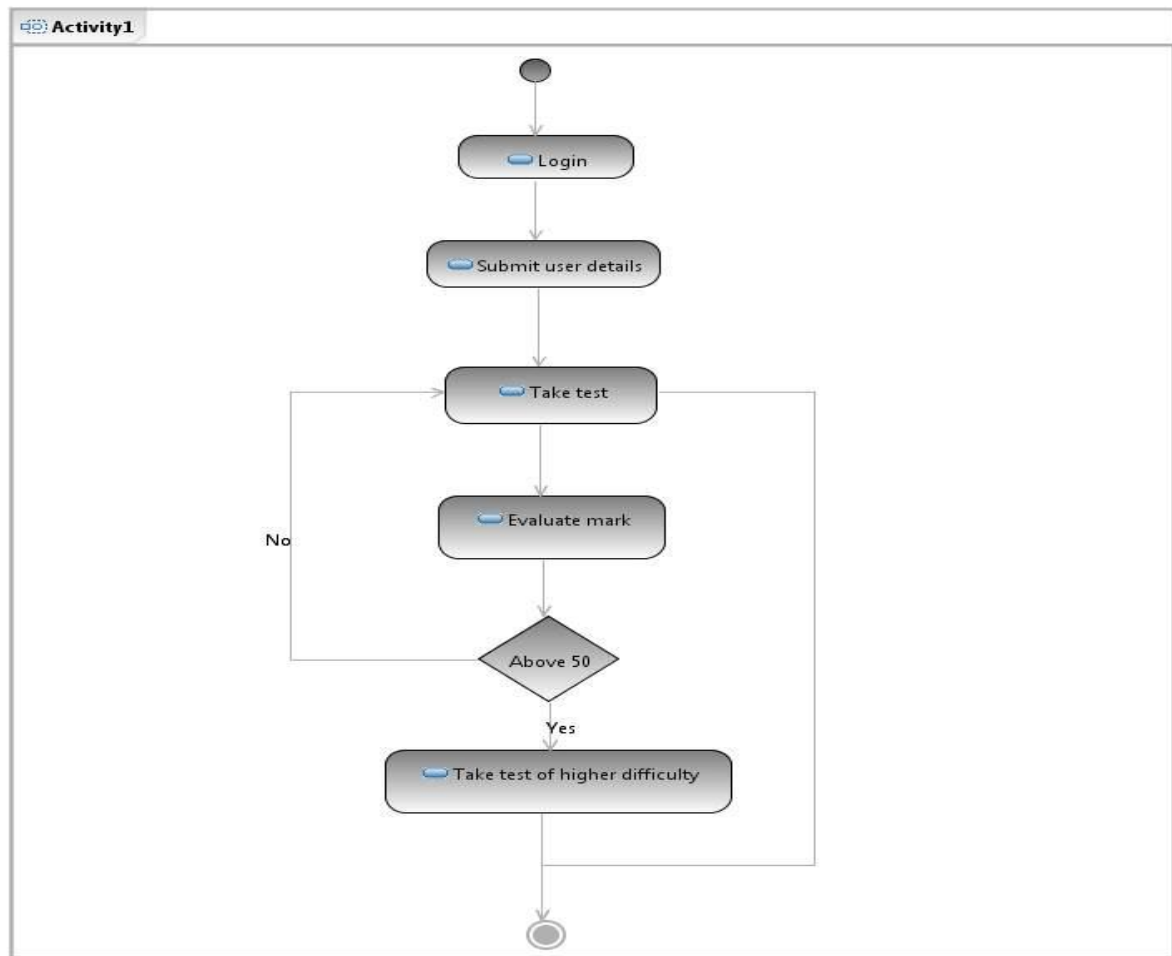


Figure 3.4 Activity diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as operations of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system. It does not show any message flow from one activity to another. Activity diagram shown in Figure 3.4 starts with a login page. When the user id and password is submitted, it goes to the next process of taking the test. After taking the test it helps in evaluating the marks. Also if the marks are above 50, higher difficulty exam is given to the user. Each activity depends on each other, as each process takes place sequentially.

3.5.3 Sequence Diagram

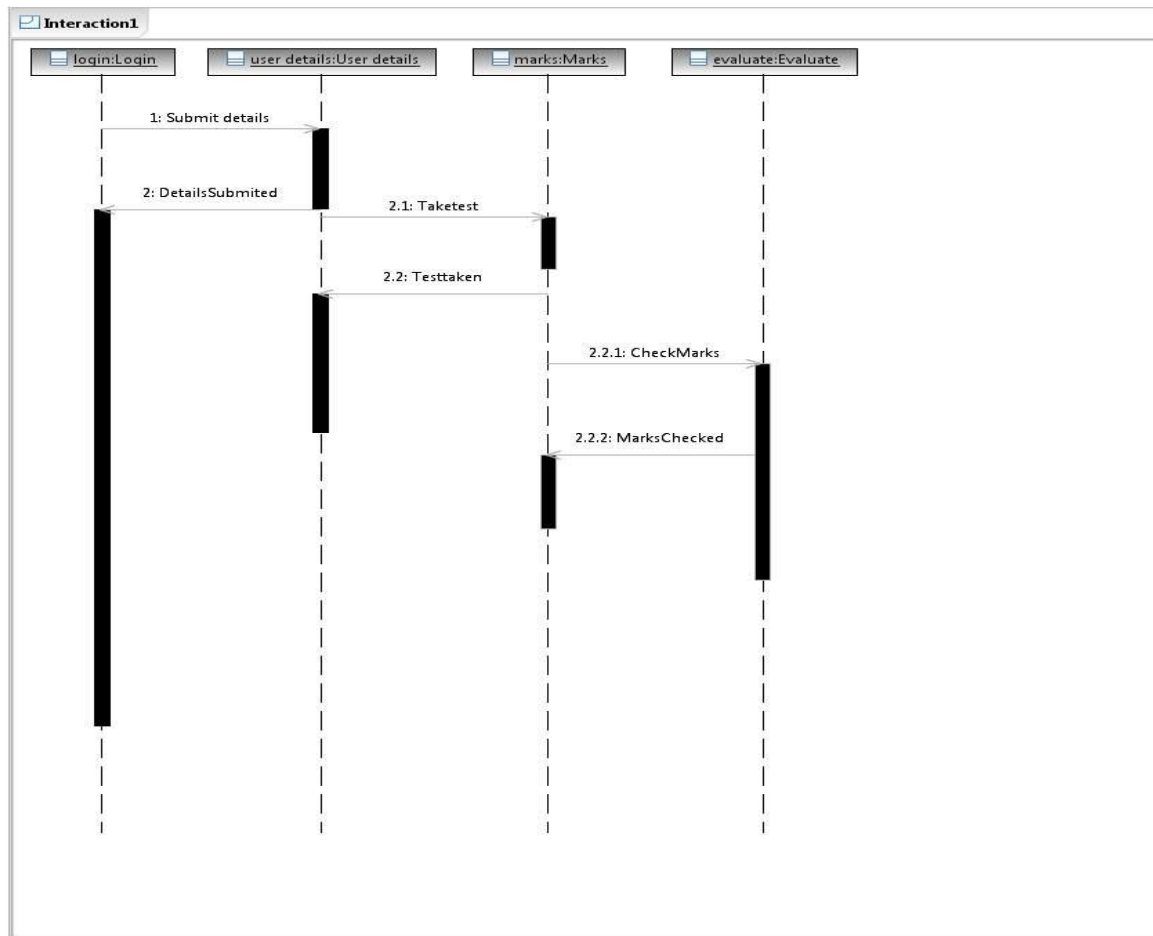


Figure 3.5 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. The sequence diagrams shown in figure 3.5 shows all the sequence process which are depicted along with the interactions between them. Initially the user gives the password and user name to the login page. Then the user is allowed to take the test. When test gets over , the results are been compiled and evaluated. Then the compiler itself makes the report and sends it to the user. The admin administrates the process and maintain the questions.

CHAPTER 4

IMPLEMENTATION AND TESTING

The online Debugging module is implemented as a user friendly website which has the facility to debug C and C++ online. This module has a unique feature of sorting the type of C program questions based on the user criteria. The adaptive function will test the user based on their performance and helps the user to improve their programming skills in that particular programming language. Then the results will be finalized and displayed on a separate screen. Then all the user logins can be seen on the admin page in which the result could be seen.

4.1 LIST OF ALGORITHM

4.1.1 Adaptive Test Algorithm

Step 1: The user input for Beginner, intermediate and expert is taken in a drop down menu.

Step 2: The conditions for sorting the programs based on the user criteria is taken.

Step 3: Now all the programs available for the test module is compiled once and made ready for adaptive function.

Step 4: The programs are categorized based on their number of lines, no of errors, and the number of functions, loops, variables, pointers, arrays and header files used.

Step 5: Now the Adaptive function will separate the programs based on these criteria and then the programs will be segregated into EASY, MEDIUM and HARD Destinations.

$$\text{Pg Weigt.} = \frac{(\text{no. of Fn} * 3) + (\text{no. of loops} * 2) + (\text{no. of ptr} * 1) + (\text{no. of arr} * 1) + (\text{no. of var} * 0.5)}{5}$$

$$\text{Total weight} = \text{No. of errors} + \text{no. of lines} + \text{pg Weight}$$

Step 6: Now the test module will send one normal level program.

Step 7: Based on the user's performance in the previous programs the adaptive function will decide whether to send easy program or hard program.

4.2 ALGORITHM DESCRIPTION

1.2.1 Flow Chart

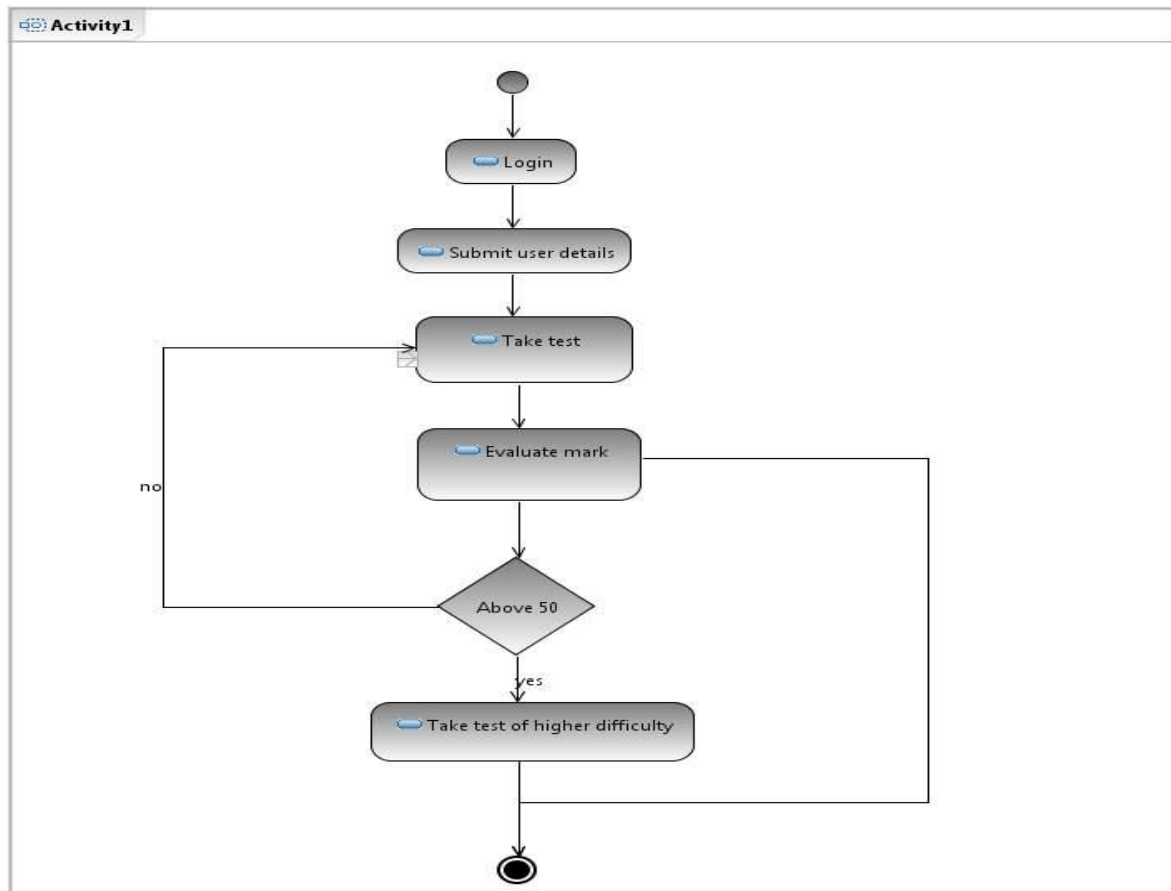


Figure 4.1 Flow chart

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields. Flow Chart shown in Figure 4.1 starts with a login page. When the user id and password is submitted, it goes to the next process if taking the test. After taking the test it helps in evaluating the marks. Also if the marks is above 50, higher difficulty exam is given to the user. Each activity depends on each other, as each process takes place sequentially.

4.3 TEST PLAN

The Online Debugging Evaluator Module is implemented online. So the testing methodologies that are used to test this module are

- Unit Testing
- Integration Texting
- Stress Testing

1.4 TEST PROCEDURES

The testing Procedures used for the following testing methodologies are explained below.

Testing procedure for unit testing:

Unit testing is the methodology used to find out the efficiency of each of the modules. This testing is individually carried out on all the modules of the project to estimate the efficiency.

Testing procedure for integration testing:

The Integration testing is done to find out the adaptability and the integrity of all the modules to perform one single process. This testing is carried out by checking all the data flow and the control flow of all the modules of the project.

Testing procedure for stress testing:

The Stress testing is done to check the load capacity of the project. The project module is hosted in a server and all the other computers are connected in a WiFi or a LAN and the module is run with minimum capacity. Then the load is gradually increased and then the load capacity at each point is judged.

4.5. TEST CASES

Test cases For Adaptive Function:

Test Case#: 1

System: web application

Designed By: Jewel and Jeffin

Executed By: Jewel and Jeffin

Short Description: registering a new user and attempting to use the same login again.

Test Case Name: new user

Design Date:20/02/2015

Execution Date:05/03/2015

Type of testing: unit testing

Pre Condition: A new user is required to enter the relevant details in all the fields such as name, user name, e-mail id, password.

Table 4.1 Test case for adaptive function

Step	Action	Input	Expected System Response	Pass/Fail
1	User logs into the web page		The various options available to the user are displayed	Pass
2	User clicks on new user option		All the necessary fields for the purpose of registration are displayed.	Pass
3	User clicks on the user name option	Name of the user	The user name is accepted	Pass
4. a.	User clicks on roll no.	Roll no. is typed.	The roll no. is accepted.	pass
4.b.	User directed to debugging module.	Correct the error program	A dialog box appears having the program for debugging.	pass
5.	After debugging the user clicks the compile button	The program is compiled and the no. of errors is displayed	A dialog box indicating the empty fields requiring an input is displayed.	Pass
6.	The mark sheet is displayed		The dialog box indicating the mark is displayed	pass

Test cases For Debugging Module:**Test Case#:** 2**System:** web application**Designed By:** Jewel and Jeffin**Executed By:** Jewel and Jeffin**Short Description:** a user is taking the debugging test.**Test Case Name:** new user**Design Date:**25/02/2015**Execution Date:**05/03/2015**Type of testing:** debugging testing

Pre Condition: A new user is required to enter the relevant details in all the fields and then take up the debugging test in the debugging module.

Table 4.2 Test case for Debugging module

Step	Action	Input	Expected System Response	Pass/Fail
1	User enters the debugging page		The various options available to the user are displayed	Pass
2	The timer is on for 2 mins		All the necessary fields for the purpose of debugging with respect to time is displayed.	Pass
3	User clicks on the program	The program corrections are made	The corrections are accepted.	Pass
4. a.	User clicks on compile button	Click on compile.	The program is compiled and the number of errors is shown.	pass
4.b.	The user is now directed to the submit option.	User clicks on submit button.	A evaluator module opens.	pass
5.	The evaluator module shows the result		The evaluator page displays the user performance analysis.	Pass

Test cases For Adaptive Function:**Test Case#:** 3**System:** web application**Designed By:** Jewel and Jeffin**Executed By:** Jewel and Jeffin**Short Description:** registering a new user and attempting to use the same login again**Test Case Name:** new user**Design Date:**28/02/2015**Execution Date:**05/03/2015**Type of testing:** integrated testing

Pre Condition: A new user is required to enter the relevant details in all the fields such as name, user name, e-mail id, password.

Table 4.3 Test case for adaptive function

Step	Action	Input	Expected System Response	Pass/Fail
1	User logs into the web page		The various options available to the user are displayed	Pass
2	User clicks on level	User chooses the level	The level is given as input	Pass
3	Compilation of programs		All the programs are compiled and the adaptive function is executed	Pass
4.	The programs are sorted based on weight		After compilation the programs are sorted based on result	pass
5.	The user is provided with a medium program	Correct the error program and compile	Now the performance is analysed and the adaptive function chooses the next program for the user based on the criteria	pass

CHAPTER 5

RESULTS AND DISCUSSIONS

- The evaluation of the debugged programs is based on the number of compilations and the time taken to complete. The result is accurate.
- The adaptive function gets the necessary inputs and then works on segregates the programs based on that. This process is done accurately.
- The result sheet will contain the total performance analysis of the user. This process is also done accurately.
- The adaptive function accurately categorizes the programs as per the algorithm.
- This project helps in getting the skill of the participant in programming. The compiler helps evaluating the skill of the person in programs.
- In this project the adaptive function will test the user based on their performance and helps the user to improve their programming skills in that particular programming language.

CHAPTER 6

CONCLUSION AND FUTURE WORK

There are many symposiums and technical events conducted in various Engineering colleges where code debugging is done. In software companies also there is a huge opportunity for online debugging system for trainees. So to meet the technical needs of software firms and engineering colleges, this website can be modified with the capacity to include more programming languages. This project can be better modified using high level languages like JAVA, Python . This system also has higher end features which will increase the effectiveness of the test module and thereby increase the efficiency of the programmers.

This design algorithm segregates the available programs in database to easy, hard and medium based on number of errors, number of characters, number of loops, number of functions, number of variables and number of statements used. This algorithm module helps to improve programming level of learner. The adaptive function will test the user based on their performance and helps the user to improve their programming skills in that particular programming language.

APPENDIX – 1

SAMPLE SOURCE CODE

ADMIN.php

```

<html>
<head>
<title>Admin</title>
</head>
<?php $pic='admin.jpg'; ?>

<body style="background-image: url('<?php echo $pic; ?>');background-repeat: no-repeat;
background-attachment: fixed; background-position: center;" >
<center>
<h1>Welcome Admin</h1>
<h2>Results of the participants</h2>
<?php

$db_host = "localhost";
$db_username = "root";
$db_pass = "";
$db_name = "jewel";

@mysql_connect("$db_host","$db_username","$db_pass") or die ("Could not connect to
MySQL");

@mysql_select_db("$db_name") or die ("No database");

//@mysql_query("CREATE DATABASE my_testingh") or die ("Could not create table");
//@mysql_query("INSERT INTO winner1 (regnumber, name, mark)
//VALUES (1001, 'fredrick',35)");
$result=@mysql_query("SELECT * FROM `winner1`");
echo '<table border="1"; cellpadding="8"; cellspacing="25">
<tr>

```

```

<th><b><p style="font-size:20px; color:red;">Reg number</th>
<th><b><p style="font-size:20px; color:red;">Name</th>
<th><b><p style="font-size:20px; color:red;">Mark</th>
</tr>';

```

```

while($row = @mysql_fetch_array($result))
{
    echo "<tr>";
    echo "<tr>";
        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['regnumber']
. "</td>";
        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['name'] .
"</td>";
        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['mark'] .
"</td>";
    echo "</tr>";
}
echo "</table>";
?>
<script language="JavaScript">

```

```

function winners()
{
document.adminis.action="top3.php";
    document.adminis.submit();

}
function clrecords()
{
document.adminis.action="trunk.php";
    document.adminis.submit();

}
</script>

```

```

        <form name="adminis">
        </br></br></br>
        <input name="winner" type="button" value="Top # 3" style="width: 69px; height:
22px" onclick="winners()">
        <input name="trunk" type="button" value="clear Records" style=" height: 22px"
onclick="clrecords()">
        </center>
</form>
</body>
</html>

```

FRONT PAGE.php

```

<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<title>Register page</title>
</head>
<script language="JavaScript">

function validateForm()
{
var userName= document.getElementById('uname').value;
var rollNo= document.getElementById('regno').value;
if( userName =="" || rollNo==" ){
    alert('Please enter User Name & Roll Number');
    return false;
}
document.forms[0].submit();
}
alert('Please Dont close the browser and try to complete within the given time or else you will
be disqualified');
</script>

```

```
<?php $pict='fir.jpg'; ?>
```

```
<body style="background-image: url('<?php echo $pict; ?>'); background-repeat: no-repeat;
background-attachment: fixed; background-position: center;background-color:black ; " >
```

```
<form action="mainpage.php" form="loginForm" id="loginForm" method="post">
```

```
</br></br></br></br></br></br></br></br></br></br></br><br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

FIRSTPAGE.php

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Language" content="en-us">
```

```
<title>Register page</title>
```

```
</head>
```

```
<script language="JavaScript">
```

```
function validateForm()
```

```
{
```

```
var userName= document.getElementById('uname').value;
```

```
var rollNo= document.getElementById('regno').value;
```

```
if( userName == "" || rollNo== "" ){
```

```
    alert('Please enter User Name & Roll Number');
```

```
    return false;
```

```
}
```

```
document.getElementById('status').innerHTML="ADAPTIVE      ALGORITHM      IS
```

```
RUNNING... PLEASE WAIT!";
```

```
document.forms[0].submit();
```



```

}
alert('Please Dont close the browser and try to complete within the given time or else you will
be disqualified');
</script>

```

```

<?php $pict='fir.jpg';
$db_host = "localhost";
$db_username = "root";
$db_pass = "";
$db_name = "grandin";

```

```

@mysql_connect("$db_host","$db_username","$db_pass") or die ("Could not connect to
MySQL");

```

```

@mysql_select_db("$db_name") or die ("No database");

```

```

@mysql_query("DELETE FROM marks");
@mysql_close();
?>

```

```

<body style="background-image: url('<?php echo $pict; ?>'); background-repeat: no-repeat;
background-attachment: fixed; background-position: center;background-color:black ; " >

```

```

<form action="mainpage.php" form="loginForm" id="loginForm" method="post">

```

```

<p id="status" style="font-color:'blue';"></p>

```

```

</br></br>

```

```

<br>

```

```

<br>

```

```

<br>

```

```

<br><br>

```

```

<br>

```


GETRESULT.php

```

<?php
// from: http://php.thedemosite.co.uk/
// version 1.2
if (isset($_POST['password'])) // if the password is set then the form has been submitted on
login.php page
{
    $dbc = dbc::instance();
    $result = $dbc->prepare("SELECT * from members where username =:username and
password =:password");
    $result->bindParam(':username', $username, PDO::PARAM_STR);
    $result->bindParam(':password', $password, PDO::PARAM_STR);
    $rows = $dbc->executeGetRows($result);
    if(count($rows))      echo      "<font      color=#008000><Center><b>**Successful
Login**</b></Center></font>";
    else echo "<font color=#ff0000><Center><b>**Failed Login**</b></Center></font>";
}
else      echo      "<font      color=#ff8000><Center><b>**No      login
attempted**</b></Center></font>";
?>

```

MAIN2.php

```

<html><font size="6" color="black"><b>
<?php
$text=$_POST["TextArea"];
$timetaken=$_POST["timetaken"];
session_start();
$rollnumber=$_SESSION['regnumber'];
$refCount = $_SESSION['noviews'];
$progcount=$_POST["progcount"];

```

```

$username=$_SESSION['user'];
$filename="$rollnumber.c";

//if($timetaken > 120000)
//{
//$_SESSION['textcont']="$text";
//$fh=fopen($filename,'w');
//fwrite($fh,$text);
//header( 'Location: main3.php' ) ;
//}
//else

$_SESSION['textcont']="$text";

$fh=fopen($filename,'w');
fwrite($fh,$text);

shell_exec("cd C:\xampp\htdocs\webpage");
$output=shell_exec("tcc $filename 2>&1 &");

echo "welcome ".$username."\n</br>";
echo " Number of compiles - ".$_SESSION['noviews']."\n</br>";
$_SESSION['noviews'] = $refCount+1;
$pos = strpos($output, "*");

if ($pos == false) {
echo "no errors!!!";

}
else
{
$stringpos=strpos($output, "*")+3;
$errorcount=substr($output, $stringpos,2);
echo"Number of error- \n".$errorcount;

```

```
}
?>
```

```
<head>
```

```
<title>DeBugging!</title>
```

```
</head>
```

```
</br>
```

```
<tr>
```

```
<td><input name="countdown" id="countdown" type="text" style="width: 35px; height:
22px;font-size:18px;font-weight: bold;"></td>
```

```
<td><b>Seconds Remaining</b></td>
```

```
</tr>
```

```
<?php $pic='main.jpg'; ?>
```

```
<body style="background-image: url('<?php echo $pic; ?>');background-repeat: no-repeat;
background-attachment: fixed; background-position: center; background-size: cover;" >
```

```
<form name="mainform2" method="post">
```

```
<br>
```

```
<textarea name="TextArea" style="width: 800px; height: 450px"><?php echo $text;
?></textarea>&nbsp;
```

```
<input type="hidden" name="progcount" id="progcount" value='<?php echo $progcount;
?>'>
```

```
<input type="hidden" name="timetaken" id="timetaken" value='<?php echo $timetaken; ?>'>
```

```
<input name="compile" type="button" value="compile" style="width: 69px; height: 22px"
onclick="comp()">
```

```
<input name="final" type="button" value="submit" style="width: 69px; height: 22px"
onclick="nextquestion()">
```

```
</font>
```

```
</form>
```

```
<script language="JavaScript">
```

```

var d=120-document.getElementById("timetaken").value;
var myVar=setInterval(function(){myTimer()},1000);
var t;
function myTimer()

{
t=d--;
if(t<=-1)
{
myStopFunction()
}
else
{
document.getElementById("countdown").value=t;
}
}
function myStopFunction()
{
    clearInterval(myVar);
    document.mainform2.action="calcResult.php";
    document.mainform2.submit();
}

function comp()
{
document.getElementById("timetaken").value=120-t;
document.mainform2.action = "main2.php";
document.mainform2.submit();
}
function nextquestion()
{
document.mainform2.action = "calcResult.php";
document.mainform2.submit();
}

```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

MAIN3.php

```
<html><font size="6" color="ffffff"><b>
```

```
</br></br></br>
```

```
<?php
```

```
session_start();
```

```
$refCount = $_SESSION['noviews'];
```

```
$rollnumber=$_SESSION['regnumber'];
```

```
$usname=$_SESSION['user'];
```

```
$loggingtime=$_SESSION['logintime'];
```

```
$timetaken= (round(microtime(true) * 1000))- $loggingtime;
```

```
if($timetaken > 1200000)
```

```
{
```

```
    header( 'Location: fail.php' ) ;
```

```
}
```

```
?>
```

```
<center>
```

```
<head>
```

```
<title>result!</title>
```

```
</head>
```

```
<?php $pic='resul.jpg'; ?>
```

```
<body style="background-image: url('<?php echo $pic; ?>');background-repeat: no-repeat;
background-attachment: fixed; background-position: center; background-size: cover;" >
```

```
<form name="mainform2" method="post">
```

```
<?php
```

```
$db_host = "localhost";
```

```
$db_username = "root";
```

```
$db_pass = "";
```

```
$db_name = "grandin";
```

```
@mysql_connect("$db_host","$db_username","$db_pass") or die ("Could not connect to MySQL");
```

```
@mysql_select_db("$db_name") or die ("No database");
```

```
//@mysql_query("CREATE DATABASE my_testingh") or die ("Could not create table");
```

```
//@mysql_query("INSERT INTO winner (regnumber, name, mark)
```

```
//VALUES (1001, 'grandin',35)")
```

```
$result=@mysql_query("SELECT * FROM `marks`");
```

```
echo '<table border="3"; cellpadding="8"; cellspacing="30">
```

```
<tr>
```

```
<th><b><p style="font-size:20px; color:red;">PROGRAM NUMBER</th>
```

```
<th><b><p style="font-size:20px; color:red;">ERROR COUNT</th>
```

```
<th><b><p style="font-size:20px; color:red;">TIME MARKS</th>
```

```
<th><b><p style="font-size:20px; color:red;">COMPILE MARKS</th>
```

```
<th><b><p style="font-size:20px; color:red;">MAIN MARKS</th>
```

```
<th><b><p style="font-size:20px; color:red;">TOTAL MARKS (100)</th>
```

```
</tr>';
```

```
while($row = @mysql_fetch_array($result))
```

```
{
```

```
    echo "<tr>";
```

```
    echo "<tr>";
```

```
        echo '    <td><b><p style="font-size:20px; color:skyblue;">' . $row['prognumber'] . "</td>";
```

```
        echo '    <td><b><p style="font-size:20px; color:skyblue;">' . $row['errorcount'] . "</td>";
```



```

        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['timemark'] .
"</td>";

        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['compmark']
. "</td>";

        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['mainmark'] .
"</td>";

        echo '<td><b><p style="font-size:20px; color:skyblue;">' . $row['total'] .
"</td>";

    echo "</tr>";
}
echo "</table>";

$averagemark=0;
$result1 = @mysql_query("SELECT AVG(total) FROM marks");
while($row1 = @mysql_fetch_array($result1))
{
    $averagemark=$row1['AVG(total)'];
}
@mysql_query("INSERT INTO winner1 (regnumber, name, mark)
VALUES ('$rollnumber','$usname','$averagemark')");
echo "Average Marks Scored: ".$averagemark;
@mysql_close();
?>
<br>
</font>
</form>

</body>
</html>

```

MAINPAGE.php

```

<?php $username=$_POST["uname"];
$regnumber=$_POST["regno"];
$progcount=$_POST["progcount"];
echo "ADAPTIVE TEST ALGORITHM FINISHED! FILES ARE PUT IN YOUR
DESKTOP!!";
session_start();
$_SESSION['noviews']=1;
$_SESSION['user']="$username";
$_SESSION['regnumber']=$regnumber;
$_SESSION['logintime']= round(microtime(true) * 1000);

$parent=$regnumber;
$path="C:\Users\Shibin\Desktop\$parent";

$pathname1=$path;
mkdir ( $pathname1,0700 );
$easy="C:\Users\Shibin\Desktop\$parent\easy";
$hard="C:\Users\Shibin\Desktop\$parent\hard";
$normal="C:\Users\Shibin\Desktop\$parent\normal";

$pathname4=$normal;
mkdir ( $pathname4,0700 );

$pathname2=$hard;
mkdir ( $pathname2,0700 );

$pathname3=$easy;
mkdir ( $pathname3,0700 );

$easycount=1;
$normalcount=1;
$hardcount=1;

```

```

for ($x=1; $x<=17; $x++)
{
    $fname=$x."_C.c";
    $output1=shell_exec("tcc $fname 2>&1 &");

    $strpos=strpos($output1, "*")+3;
    $errcount=substr($output1, $strpos,2);
    $program = file_get_contents($fname);
    $nlines=strlen($program);

    $text= $program;

    $loop= substr_count($text, 'for');

    $while= substr_count($text, 'while');

    $var= substr_count($text, 'int');
    $main=substr_count($text, 'main');

    $if=substr_count($text, 'if');

    $switch=substr_count($text, 'switch');

    $ashstick=substr_count($text, '*');

    $array=substr_count($text, '[');

    $total=$loop+$while+$var+$main+$if+$switch+$ashstick+$array;
    //echo $total."<br>";

```

```

if($errcount <= 1 && $nlines <= 300 && $total <=5)
{

//echo"easy<br>";
$adpfname=$easy."/".$easycount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);
$easycount++;
}
elseif($errcount <= 1 && $nlines <= 400 && $total <=10)
{

//echo"normal<br>";
$adpfname=$normal."/".$normalcount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);
$normalcount++;
}
elseif($errcount <= 1 && $nlines <= 600 && $total <=5)
{

//echo"hard<br>";
$adpfname=$hard."/".$hardcount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);
$hardcount++;
}
elseif($errcount <=2 && $nlines <= 200 && $total <=10)
{

//echo"normal<br>";
$adpfname=$normal."/".$normalcount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);

```

```

$normalcount++;
}
elseif($errcount <= 2 && $nlines <= 400 && $total <=10)
{

//echo"normal<br>";
$adpfname=$normal."/".$normalcount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);
$normalcount++;
}
elseif($errcount <= 2 && $nlines <= 600 && $total <=15)
{

//echo"hard<br>";
$adpfname=$hard."/".$hardcount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);
$hardcount++;
}
elseif($errcount <=3 && $nlines > 10 && $total <=15)
{

//echo"hard<br>";
$adpfname=$hard."/".$hardcount."_C.c";
$fh=fopen($adpfname,'w');
fwrite($fh, $program);
$hardcount++;
}
}
?>

<form action="mainpage1.php" name="mainform" id="mainform" method="post">

```

```

<input type="hidden" name="progcount" id="progcount" value='<?php echo $progcount;
?>'>
</form>
<script>
document.mainform.submit();
</script>
</body>
</html>

```

MAINPAGE1.php

```

<?php

$progcount=$_POST["progcount"];
session_start();
$username=$_SESSION['user'];
$regnumber=$_SESSION['regnumber'];
$_SESSION['noviews']=1;

$parent=$regnumber;

$easy="C:\Users\Shibin\Desktop\$parent\easy";
$hard="C:\Users\Shibin\Desktop\$parent\hard";
$normal="C:\Users\Shibin\Desktop\$parent\normal";

$previouserrorcount=-1;
$db_host = "localhost";
$db_username = "root";
$db_pass = "";
$db_name = "grandin";

@mysql_connect("$db_host","$db_username","$db_pass") or die ("Could not connect to
MySQL");

```

```
@mysql_select_db("$db_name") or die ("No database");
```

```
$result=@mysql_query("SELECT errorcount FROM marks WHERE prognumber=(SELECT  
MAX(prognumber) FROM marks)");
```

```
while($row = @mysql_fetch_array($result))  
{
```

TOP.php

```
<html>
```

```
<head>
```

```
<title>Winners</title>
```

```
</head>
```

```
<?php $pic='winner.jpg'; ?>
```

```
<body style="background-image: url('<?php echo $pic; ?>');background-repeat: no-repeat;  
background-attachment: fixed; background-position: center;" >
```

```
<font size="4" color="black"><b>
```

```
<center>
```

```
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
```

```
<?php
```

```
$db_host = "localhost";
```

```
$db_username = "root";
```

```
$db_pass = "";
```

```
$db_name = "grandin";
```

```
@mysql_connect("$db_host","$db_username","$db_pass") or die ("Could not connect to  
MySQL");
```

```
@mysql_select_db("$db_name") or die ("No database");
```

```
$top3=@mysql_query("SELECT `regnumber`, `name`, `mark` FROM `winner1` WHERE 1
```

```

ORDER BY `winner1`.`mark` DESC LIMIT 3");
echo '<table cellpadding="8"; cellspacing="25">
<tr>
<th><b><p style="font-size:20px; color:red;">Reg number</th>
<th><b><p style="font-size:20px; color:red;">Name</th>
<th><b><p style="font-size:20px; color:red;">Mark</th>
</tr>';
while($winner = @mysql_fetch_array($top3))
{
echo "<tr>";
echo '<td><b><p style="font-size:20px; color:skyblue;">' . $winner['regnumber'] . "</td>";
echo '<td><b><p style="font-size:20px; color:skyblue;">' . $winner['name'] . "</td>";
echo '<td><b><p style="font-size:20px; color:skyblue;">' . $winner['mark'] . "</td>";
echo "</tr>";
}
echo "</table>";
?>

</b></font>

<script language="JavaScript">

function back()
{
document.adback.action="admin.php";
document.adback.submit();

}

</script>
<form name="adback">
<input name="backad" type="button" value="Back" style=" height: 22px" onclick="back()">
</center>
</body>
</html>

```



```
{  
document.delt.action="admin.php";  
document.delt.submit();  
  
}  
</script>  
<form name="delt">  
<input name="trunk" type="button" value="Back" style=" height: 22px" onclick="back()">  
</center>  
</form>  
</body>  
</html>
```

APPENDIX – 2

SCREENSHOTS

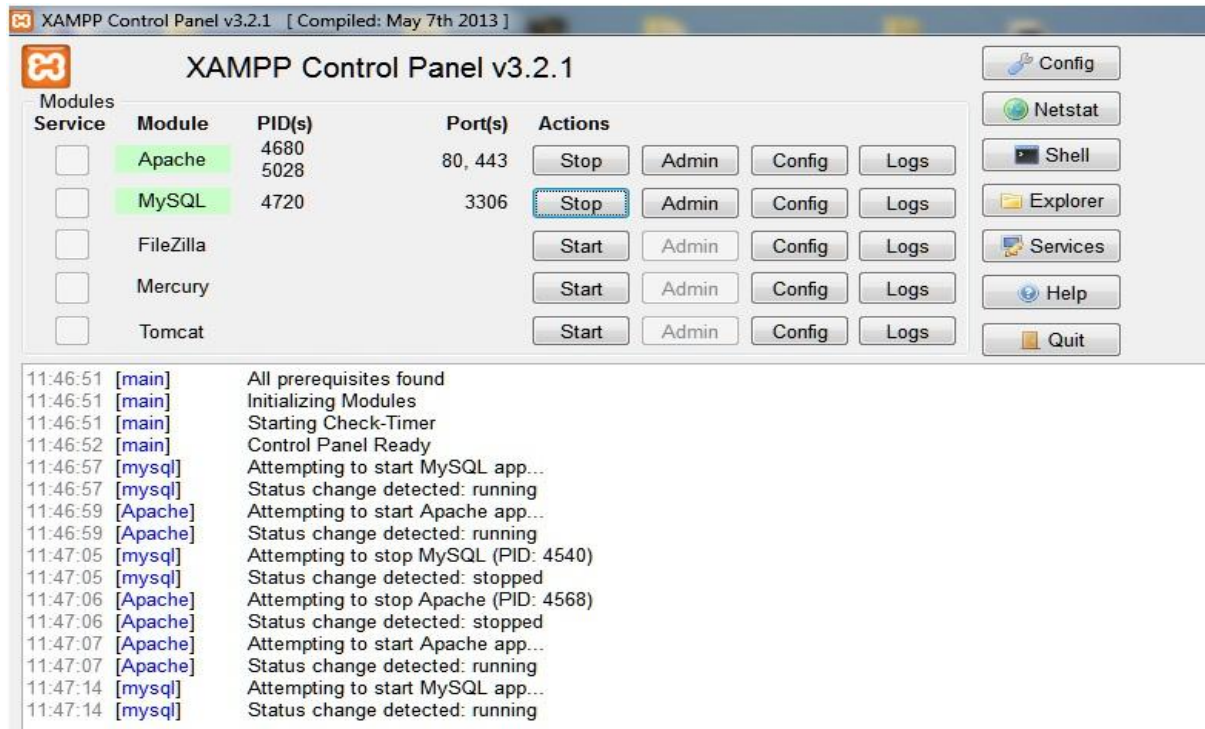


Figure A2.1 XAMPP control Panel

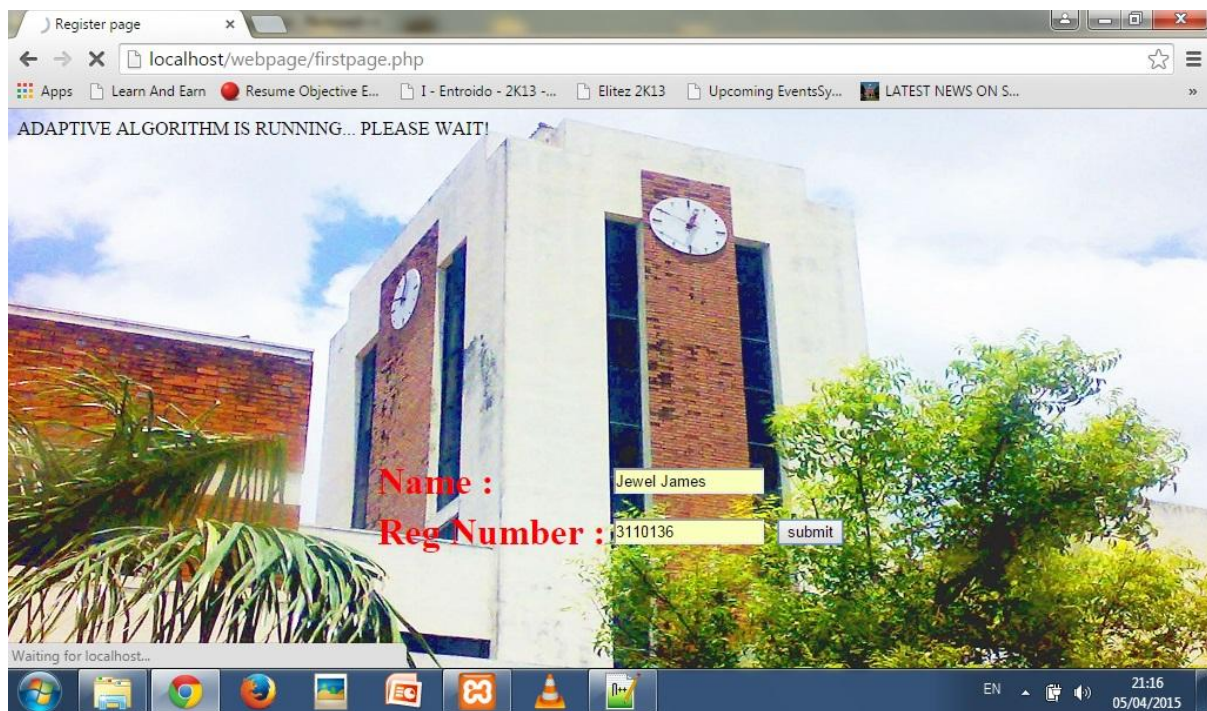


Figure A2.2 Register Page

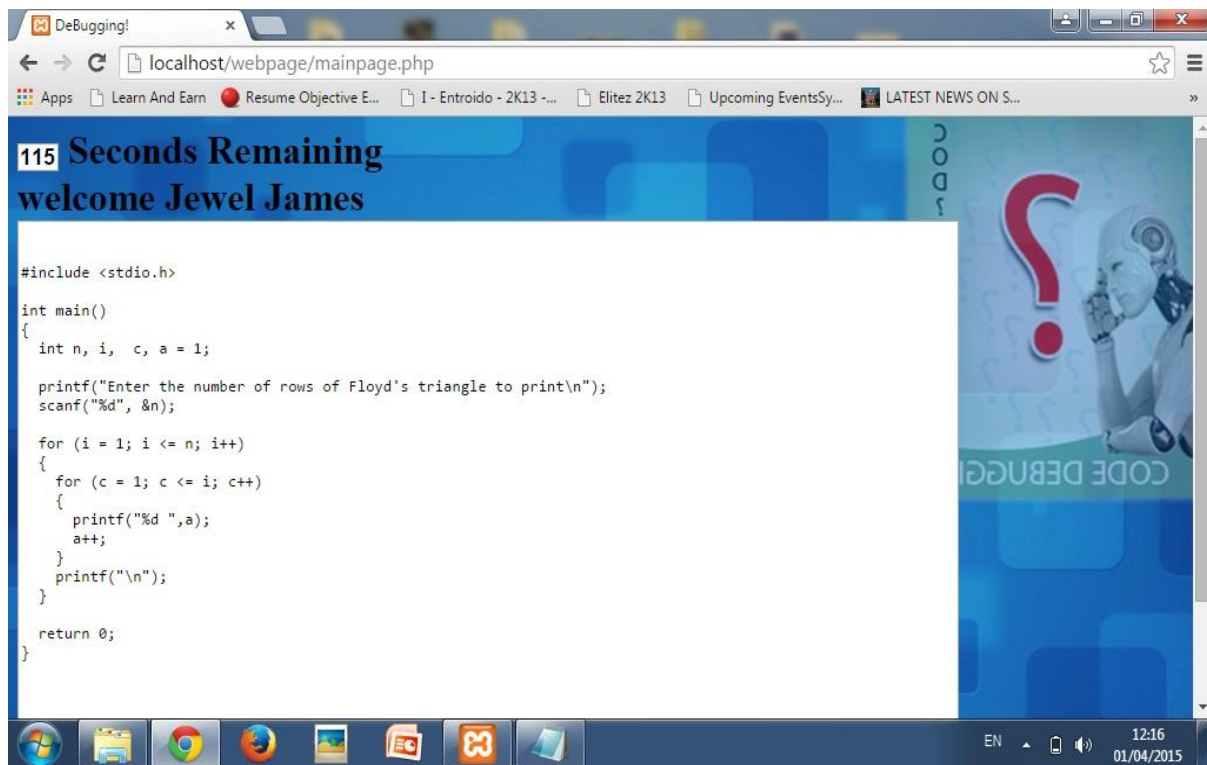


Figure A2.3 Test page 1

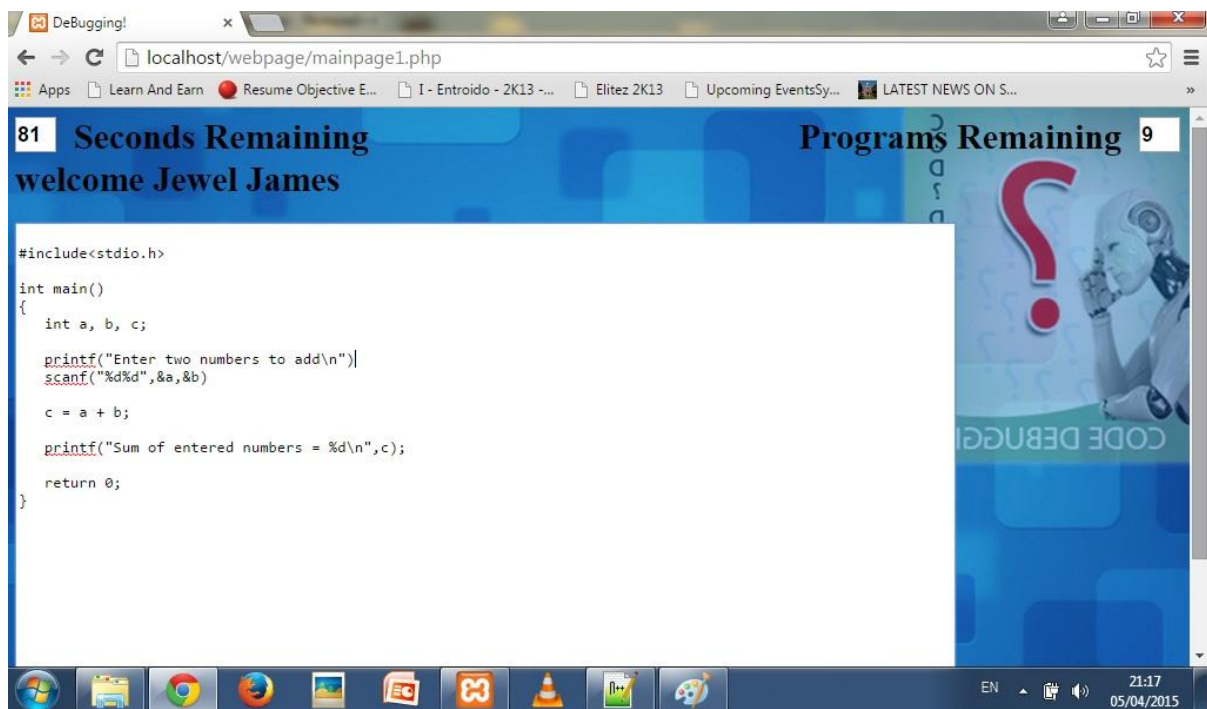


Figure A2.4 Test Page 2

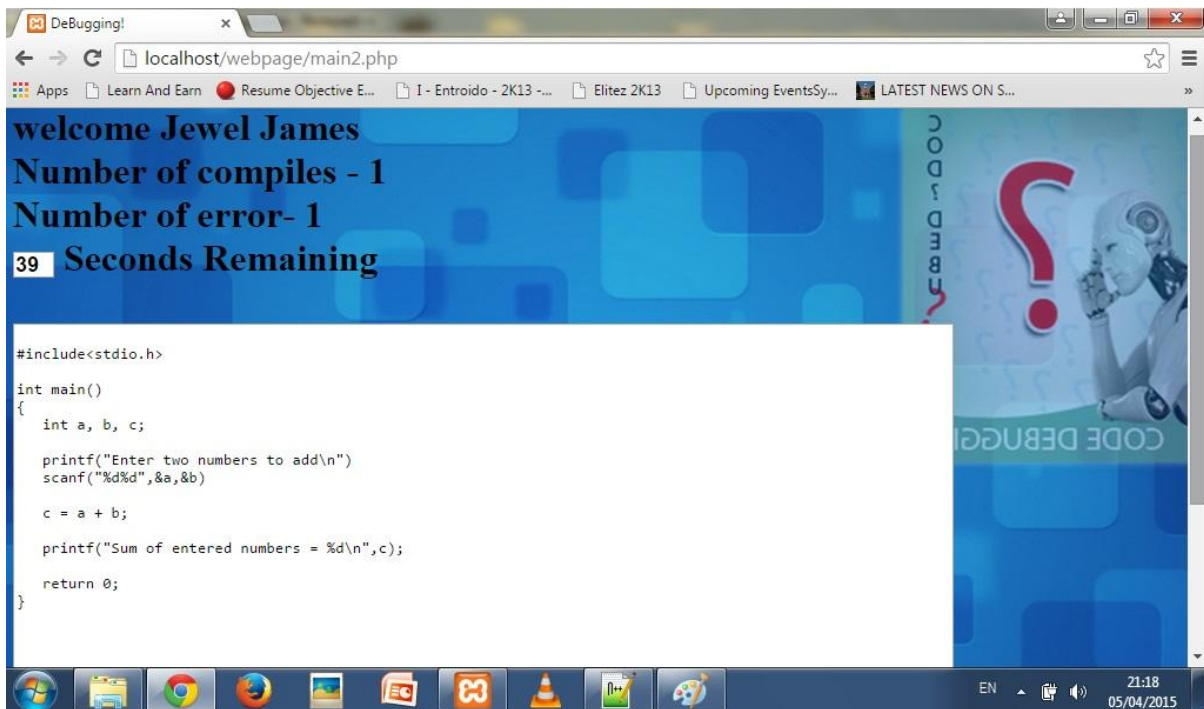


Figure A2.5 Test page 3

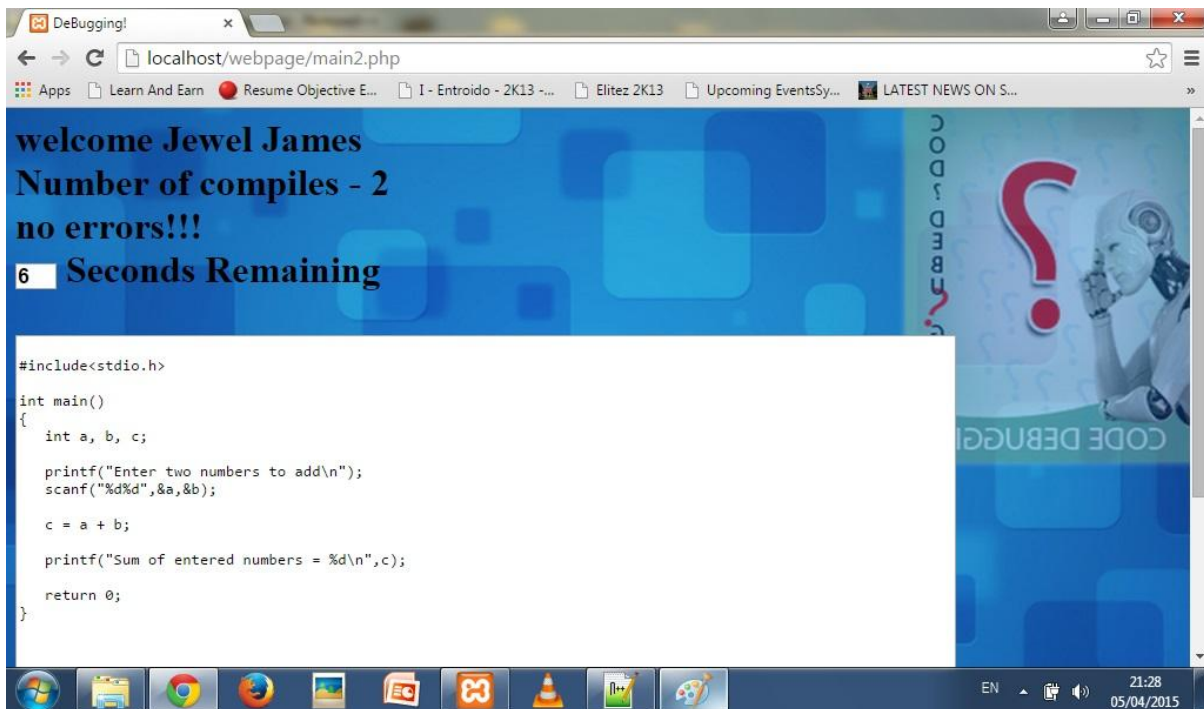


Figure A2.6 Test Page 4



Figure A2.7 Report Page



Figure A2.8 Results of the participants

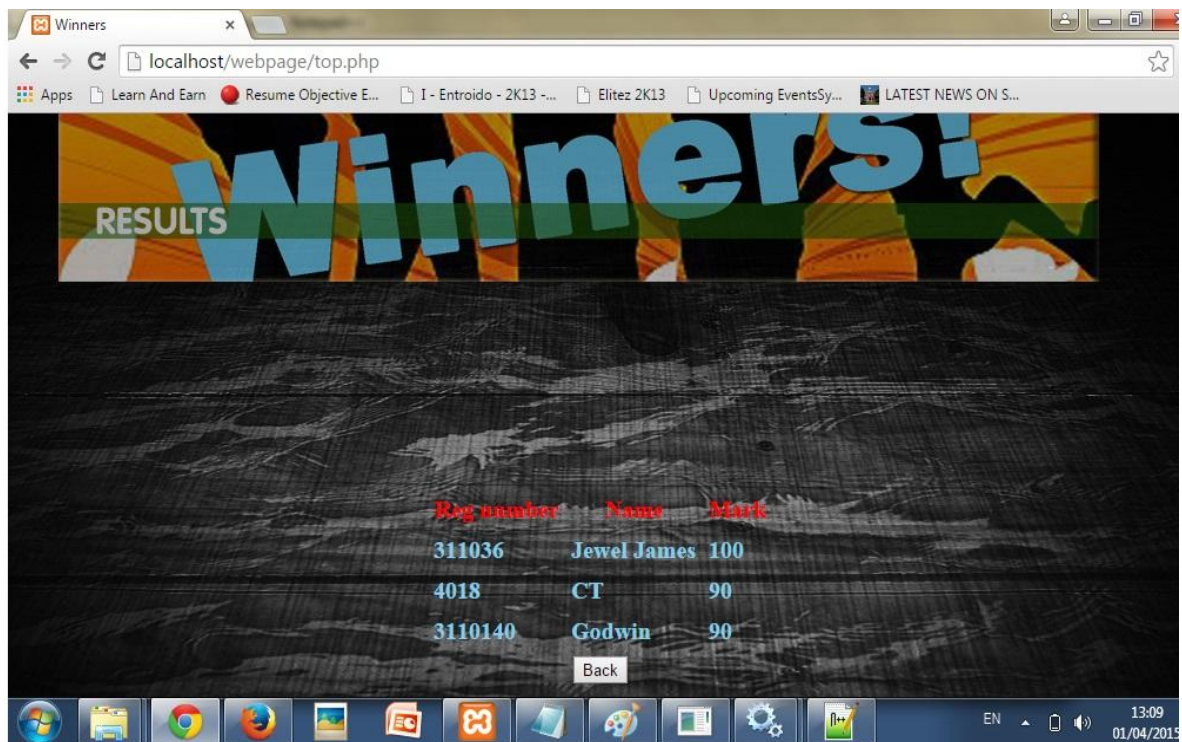


Figure A2.6 Final Report Page

REFERENCES

1. Armen C., and Morelli, R. (2005) 'Computer science: Teaching about the risk of electronic technology', ACM ITiCSE.
2. Andrea Zisman, George Spanoudakis, James Dooley and Igor Siveroni. (2009) 'Proactive and reactive Runtime Service Discovery: A Framework and its Evaluation', IEEE.
3. Mahbub, K., Spanoudakis, G. and Zisman, A. (2011) 'Runtime service discovery', Automated Software Engineering.
4. May Chan, K.S. (2011) 'Self-healing cycle for web service composition'.
5. Rebhi Baraka*, Olga Caprotti, and Wolfgang Schreiner,(2014) ' A Web Registry for Publishing and Discovering Mathematical Services'.
6. Sheng, Q.Z., Benatallah, B., Maamar, Z. and Ngu, A.H.H. (2007) 'Configurable composition and adaptive provisioning of web services', IEEE.
7. Stallings W. (2003) 'Network security, principles and practice', Third edition, pp.67-68 and 317-375, Prentice Hall.