UNIVERSITY OF PLYMOUTH

NSBM Green University

Faculty of Computing

Web site for managing garbage collections

PUSL2020

Software Development Tool and Practices

Module Lectures : Dr. Rasika Ranaweera
Ms. Pavithra Subhashini

Acknowledgement

We would like to express our deep gratitude to everyone who contributed to the successful completion of this project. We are indebted to numerous individuals and organizations who helped us in various ways, and without whom this project would not have been possible.

First and foremost, we would like to thank our project supervisors Dr.Rasika Ranaweera and Ms. Pavithra Subhashini who provide us invaluable guidance, support, and feedback throughout the project. Their insights and knowledge have been instrumental in shaping the direction and scope of the project.

We would also like to thank our team members, who worked tirelessly and cooperatively to ensure the project's success. Their commitment, dedication, and hard work played a crucial role in completing the project.

Finally, we would like to thank all the individuals and organizations who have contributed to the project, including the various online resources, libraries, and software tools that were used. Their contributions have been invaluable in helping us achieve the project's objectives.

In conclusion, we recognize and acknowledge the contributions of all those who played a part in the success of this project. Thank you for your support, encouragement, and collaboration, and for helping us make a positive impact on the environment and society.

Table of contents

1. Acknowledgement.....2
2. Content.....3
3. Introduction.....4
4. Testing Types.....5
    4.1. Unit Testing
    4.2. Integration Testing
    4.3. Functional Testing
    4.4. Role of any mock object
5. Test and validation metrics.....22
6. Use of tools and techniques.....23
7. Working screen shots.....24
8. Contribution.....28

List of figures

9. Figure 1.....24
10. Figure 2.....24
11. Figure 3.....25
12. Figure 4.....25
13. Figure 5.....26
14. Figure 6.....26
15. Figure 7.....27
16. Figure 8.....27

List of Tables

17. Table 1.....7
18. Table 2.....8
19. Table 3.....9
20. Table 4.....10
21. Table 5.....11
22. Table 6.....12
23. Table 7.....13
24. Table 8.....14
25. Table 9.....15
26. Table 10.....16
27. Table 11.....17
28. Table 12.....18
29. Table 13.....19
30. Table 14.....20
31. Table 15.....21

3. Introduction:

The increasing concern for environmental conservation has become a global priority in recent years. As a result, many organizations and individuals are taking steps towards promoting environmental cleanliness and sustainability. In response to this global challenge, our team has developed a web-based platform aimed at promoting environmental cleanliness by enabling users to report garbage locations and allowing garbage collecting staff to efficiently clean them.

Our platform provides an interactive and user-friendly interface for users to report garbage locations in their locality. The platform also enables the garbage collecting staff to quickly access and clean these locations, which helps to keep the environment clean and healthy. Additionally, the platform provides a centralized system for managing the entire process, from user reporting to garbage collection, which makes the process more efficient and streamlined.

In this project report, we will provide a detailed description of our web-based platform and its functionalities. We will discuss the methodology used for the development of the platform, including the tools and technologies used, as well as the challenges faced during the development process. We will also describe the test strategy adopted for the project, including the types of tests performed and the results obtained.

Finally, we will conclude the report by summarizing the key achievements of the project and discussing the potential for future improvements and developments.

4. Testing types

4.1 Unit Testing

Unit testing is a software testing technique where individual units or components of a software application are tested in isolation from the rest of the application.

- So here we must break down our entire project into smaller pieces until it becomes small components and then we must check every individual component separately. For doing this we follow below steps.
- Our team analyze the project and identify process of every php page according to our web project and then we mark most essential part which are most embrace with user.
- Then we must write test cases for every part separately.
- Finally, we have to very whether it actual result match with expected result.

  For an example we break down registration form into small pieces and tested it separately. Here are the test cases which we used for verifying whether it is working or not.

Example 01

Registration form validation:

Testing Function: Check whether register form input filed working properly.
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RVD_001	Check whether all the input filed of Registration form working properly by adding text data separately	User must click registration button which is available on index.php page. Add relevant data for input field. Click submit.	Add only email into email field with incorrect format	Email:supportgmail.com Username:- Password - Role -	When click submit button the pop up load and it must show Please include an '@' in the email address.	It loads the pop up menu and it shows Please include an '@' in the email address.	Pass

Table 1 Registration form validation(Unit testing)

Registration form validation:

Testing Function: Check whether register form input filed working properly.
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RVD_001	Check whether all the input filed of Registration form working properly by adding text data separately	User must click registration button which is available on index.php page. Add relevant data for input field. Click submit.	Add email and Username.	Email:support@gmail.com Username:-support_amila Password - Role -	When click submit button the pop up load and it must show Password must contain at least one digit, one lowercase letter, one uppercase letter, and be at least 8 characters long	It loads the pop up menu and it shows Password must contain at least one digit, one lowercase letter, one uppercase letter, and be at least 8 characters long	Pass

Table 2 Registration form validation(Unit testing)

Registration form validation:

Testing Function: Check whether register form input filed working properly.
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RVD_001	Check whether all the input filed of Registration form working properly by adding text data separately	User must click registration button which is available on index.php page. Add relevant data for input field. Click submit.	Add email and Username and Password properly. Also add username which is already Iin database	Email:support@gmail.com Username:-support_amila Password -Support@123 Role -Admin	When click submit button then it must shows messages such as **Username already exist**	When click submit button then it shows messages such as **Username already exist**	Pass

Table 3 Registration form validation(Unit testing)

4.2 Integration Testing

Integration testing is a software testing technique that tests the integration between two or more components or subsystems of a software application.

- Here we try to combine two or more-unit testing components together. After we must run test cases for each of them.
- Finally, we must very whether its actual result match with expected result.

Here is an example for integrated testing. We combine all of the unit which we used in unit testing previously and now checking entire form to verify whether it is working or not.

Example 01

Registration form validation:

Testing Function: Check whether register form input filed working properly and pass data into database successfully.
Testing type: Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RVD_001	Check whether all the input filed of Registration form working properly by adding text data separately	User must click registration button which is available on index.php page. Add relevant data for input field. Click submit.	Add the all the relevant informations into registration form	Email:support@gmail.com Username:-support_amila Password -Support@123 Role -Admin	When click submit button then it must shows messages such as **Registration successful.**	When click submit button then it shows messages such as **Registration successful.**	Pass

Table 4 Registration form validation(Integrated testing)

4.3 Functional Testing

User account creation:

Testing Function: Stored user account details in database
Testing type: Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
UAC_001	Verify whether registration form allow to create user accounts for users which are available in different roles	User must connect both frontend and backend pages together. Backend page must be connected ith database. Database must include relevant tables with relevant field with correct structure.	User must click registration button which is available on index.php page. Add relevant data for input field. Click submit.	Email:amalperera@gmail.com Username:gtf_amalp Password:amal123 Role:gtf_member	Registration successful message must be displayed	It displayed Registration successful message.	Pass

Table 5 User account creation

Article creation:

Testing Function: Add article into database
Testing type: Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
ART_001	Verify whether admin allows to add article into database	User must connect both frontend and backend pages together. Backend page must be connected with database. Database must include relevant tables with relevant field with correct structure.	Admin must access into admin dashboard using admin credentials. Add relevant data for input field. Click submit.	Title: Description: Image:	Data inserted successfully message must be displayed	It displayed Data inserted successfully message.	Pass

Table 6 Article creation

Location marking:

Testing Function: Stored GTF member reporting details in database
Testing type: Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RPT_001	Verify whether gtf member allows to add details into database	User must connect both frontend and backend pages together. Backend page must be connected ith database. Database must include relevant tables with relevant field with correct structure. Bing map must be integrated into frontend page.	User must log as gtf_member Add relevant data for input field. Click submit.	Reporter Name Report Date Report Time Nearest City Name District Garbage Type Priority Level	New record created successfully message must be displayed	It displayed New record created successfully message.	Pass

Table 7 Location marking

Report approval_01:

Testing Function: Mark GTF member reports as Approved or Rejected
Testing type: Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RPT_002	Verify whether approve button correctly or not	User must connect both frontend and backend pages together. Backend page must be connected with database. Database must include relevant tables with relevant field with correct structure.	Green Captain must access into system using his/her credentials. Add Reason and Click Approve	Reason	Input field and both approve and reject button must be disable. Garbadge_reports table status must shows Approved	Input field and both approve and reject button are disable. Garbadge_reports table status shows Approved	Pass

Table 8 Report approval_01

Report approval_02:

Testing Function: Mark GTF member reports as Approved or Rejected
Testing type: Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RPT_003	Verify whether reject button work correctly or not	User must connect both frontend and backend pages together. Backend page must be connected with database. Database must include relevant tables with relevant field with correct structure.	Green Captain must access log into system using his/her credentials. Add Reason and Click Reject	Reason	Input field and both approve and reject button must be disable. Garbadge_reports table status must shows Rejected	Input field and both approve and reject button are disable. Garbadge_reports table status shows Rejected	Pass

Table 9 Report approval_02

Login User Validation:

Testing Function: Add incorrect details into input field and check its behavior.
Testing type: Functional
Test Case Designed By: WHP Fernando
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
UVD_001	Check whether user allows to add incorrect login details and access into system	User must connect both frontend and backend pages together. Backend page must relate to database. Database must include relevant tables with relevant field with correct structure.	User must click login button which is available on index.php page. Add sample(incorrect) data relevant data for input field. Click submit.	Email:abc@gmail.com Username:gtf_abc Password:abc123 Role:gtf_member	Unknown user role message must be displayed	It displayed. Unknown user role message.	Pass

Table 10 Login User Validation

Registration User Validation:

Testing Function: Leave empty input field and check its behavior.
Testing type: Functional
Test Case Designed By: WHP Fernando
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RVD_001	Check whether user allows to leave registration form fields without adding anything.	User must connect both frontend and backend pages together. Backend page must relate to database.	User must click Register button which is available on index.php page. Add sample(incorrect) data relevant data for input field. Click submit.	Email: Username: Password:abc123 Role:gtf_member	Complete rest of fields message must be displayed	It displayed Complete rest of fields.	Pass

Table 11 Registration User Validation

Password Field Validation:

Testing Function: Add incorrect data into password fields and check its behavior.
Testing type: Functional
Test Case Designed By: WHP Fernando
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
RVD_001	Check whether user allows to add password excludes its minimum requirements.	User must write proper coding using java script for adding password pattern.	User must click Register button which is available on index.php page. Add sample(incorrect) data password field Click submit.	Email:abc@gmail.com Username:gtf_abc Password:abc123 Role:gtf_member	Pop up menu must shows password minimum requirement such as Password must contain at least one digit, one lowercase letter, one uppercase letter, and be at least 8 characters long	Pop up menu loaded and it shows Password must contain at least one digit, one lowercase letter, one uppercase letter, and be at least 8 characters long	Pass

Table 12 Password Field Validation

Password Field data encryption:

Testing Function: Add password into password field and check whether it is visible on database as it is.
Testing type: Functional
Test Case Designed By: EANP Alexander
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
PFE_001	Check whether password is visible without encrypting in database	User must write proper coding encryption method.	User must click Register button which is available on index.php page. Add sample data into input fields Click submit.	Email:abc@gmail.com Username:gft_abc Password:Abc@123.com Role:gtf_member	Once submitted Password must be saves as encrypted one in database	Password saved as encrypted one in database	Pass

Table 13 Password Field data encryption

Navigation bar behavior:

Testing Function: Check whether navigation bar increase its size when mouse is scrolling top to bottom
Testing type: non-Functional
Test Case Designed By: Pasindu Sandamal
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
NVB_001	Check whether navigation bar increase its size when mouse is scrolling top to bottom	User must write proper coding for increase its size when mouse is scrolling top to bottom using java script.	User need to run website and hold in home page.	-	When webpage scroll top to bottom navigation bar size of navigation bar must be increased.	When webpage scroll top to bottom navigation bar size of navigation bar is increased.	Pass

Table 14 Navigation bar behavior

Web page responsiveness:

Testing Function: When decrease the screen size then check whether content adjusts according it
Testing Type : Non - Functional
Test Case Designed By: FF Ahamad
Test Priority (High / Medium / Low): High

Test Case ID	Test Description	Prerequisite	Test Procedure	Input Data	Expected Result	Actual Result	Status
WPR_001	When decrease the screen size then check whether content adjusts according to it	User must use proper framework which provide responsive features using some tags such as container and container-fluid. Or else the user must write media queries .	Click right button of mouse since home page and then click inspect. After click toggle switch emulation. Finally increase and decrease size and check screen behavior	-	Content must not be overlap each other.	Content did not overlap each other.	Pass

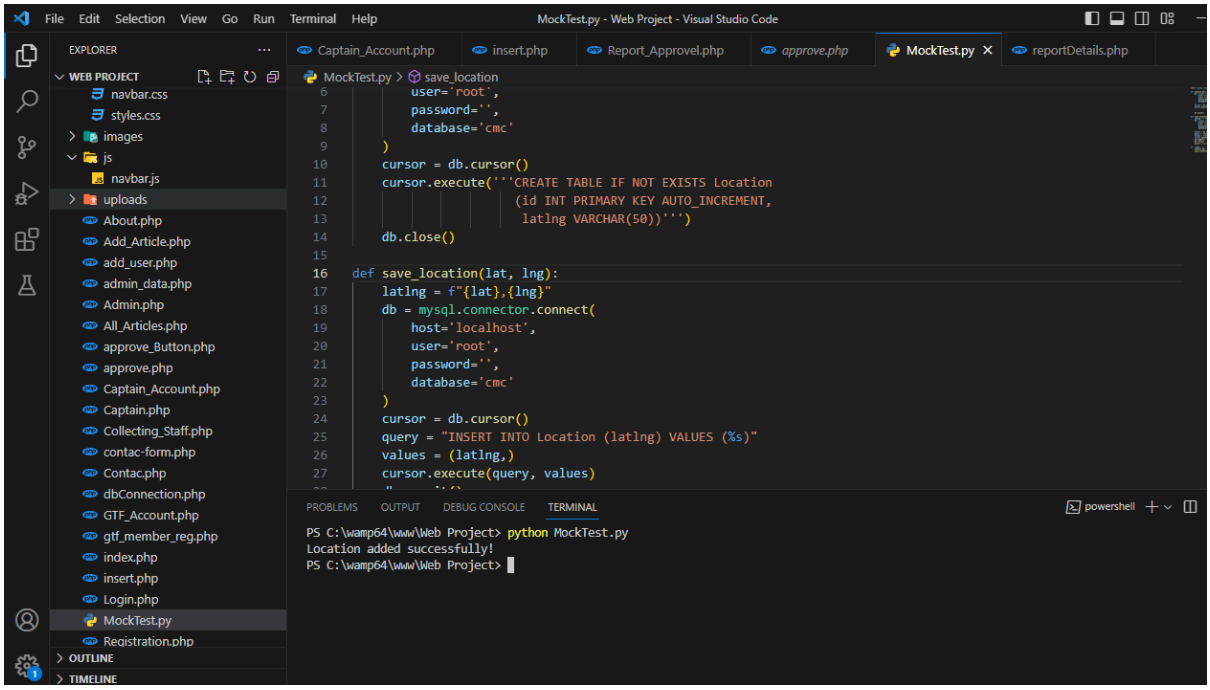
Table 15 Web page responsiveness

4.4 Role of mock test

Here we did a mock test using python. We pass sample values into the garbage location table. After running python.py then we could see data is added into table columns without any issue.

We did it using the following way.

- Install Paython into VS Code.
- Install following librires.
- pip install.
- Then we used below code.
- Finally we execute Mocktest.py file like this. python MockTest.py.
- Then it shows below result which means Location added successfully.



Mock test result

Tools and Techniques

Tools

- WAMP Server: We used WAMP Server, a Windows web development environment, to create a local server on our computer for developing and testing the website. WAMP Server includes Apache, MySQL, and PHP, which allowed us to easily create a testing environment for our website.
- Visual Studio Code: We used Visual Studio Code, a popular text editor, to write and edit our code.

Languages

- PHP: We used PHP as the primary scripting language for building the website. We used PHP to handle user input, connect to the database, and process data on the server-side.
- HTML and CSS: We used HTML and CSS to build the structure and styling of the website. We used Bootstrap, a popular CSS framework, to make the website responsive and mobile-friendly.
- Python: We used Python to create mock test data for our website. Python is a popular programming language that is well-suited for data manipulation and analysis.

Databases

- MySQL Database: We used MySQL, a popular open-source relational database management system, to store and manage data for our website. We used a combination of tables to store user information, garbage location reports, and other relevant data.

Screenshots of our project

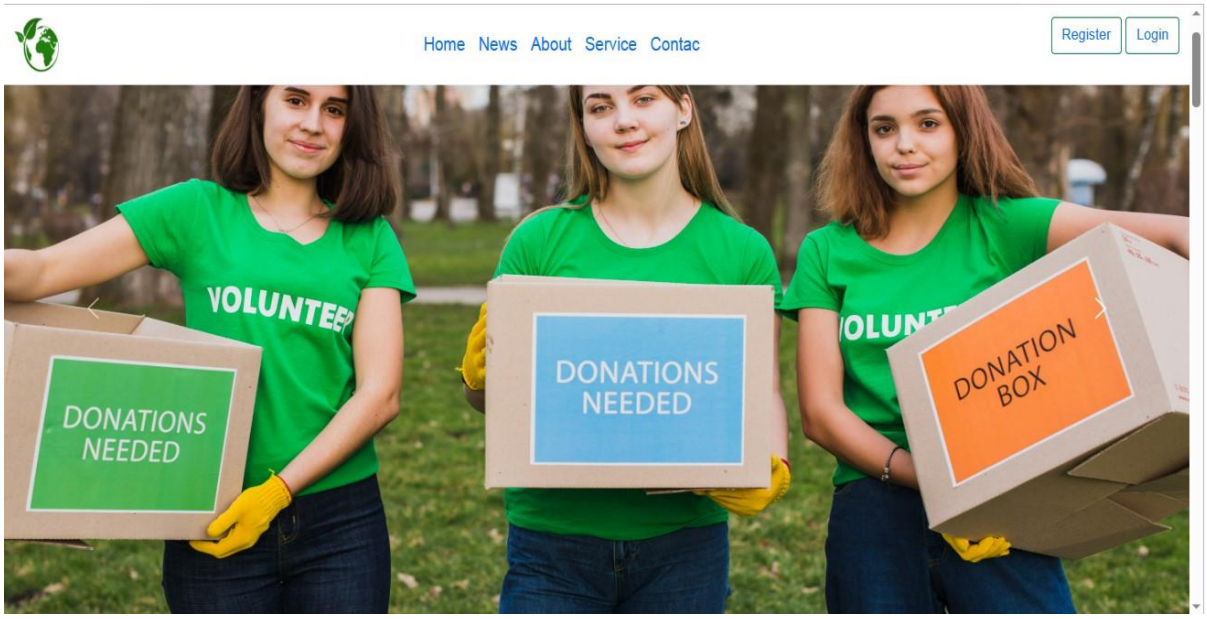


Figure 1 Home Screen of web page

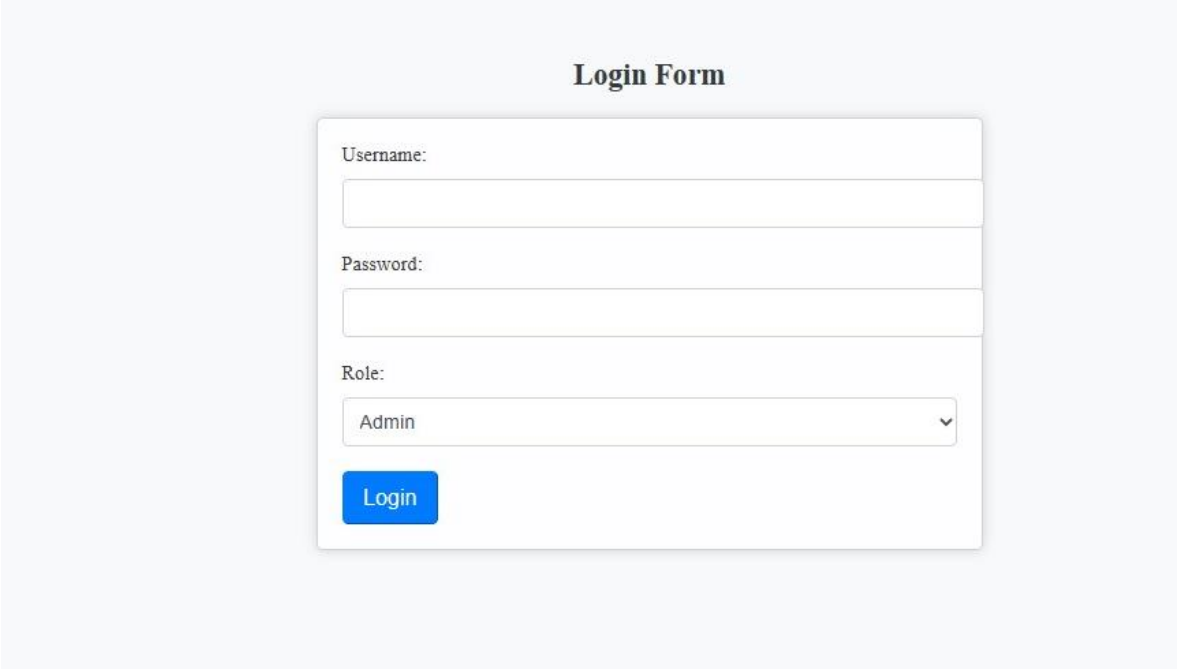


Figure 2 Login form of web page

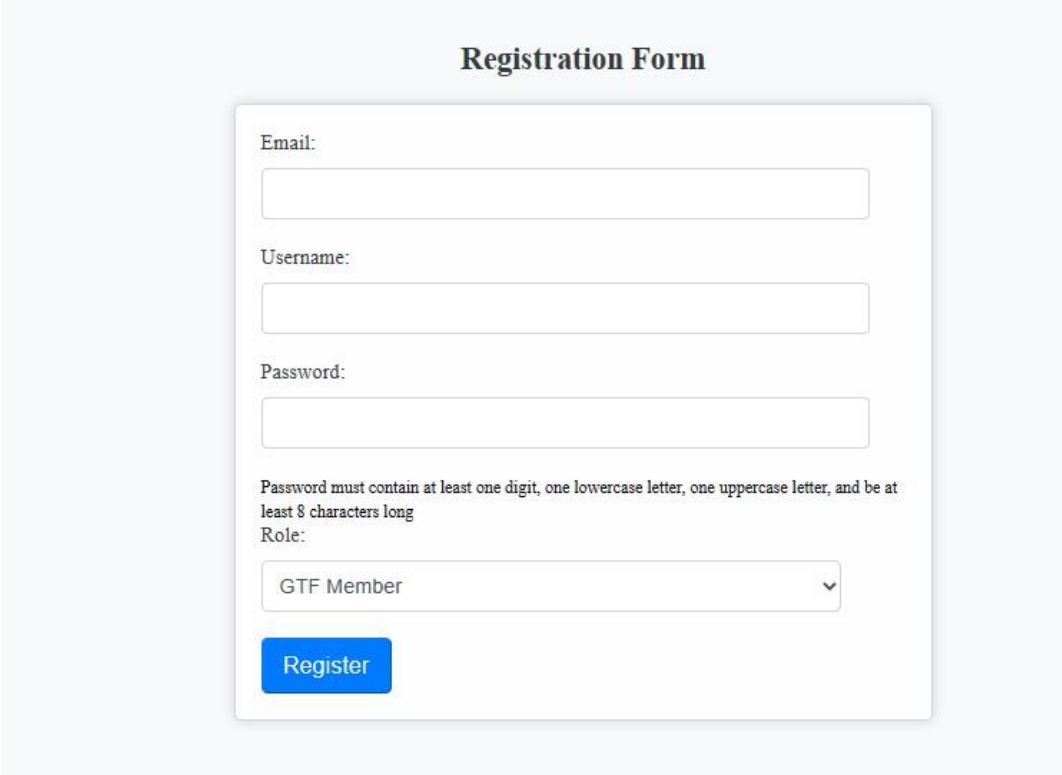


Figure 3 Registration Page

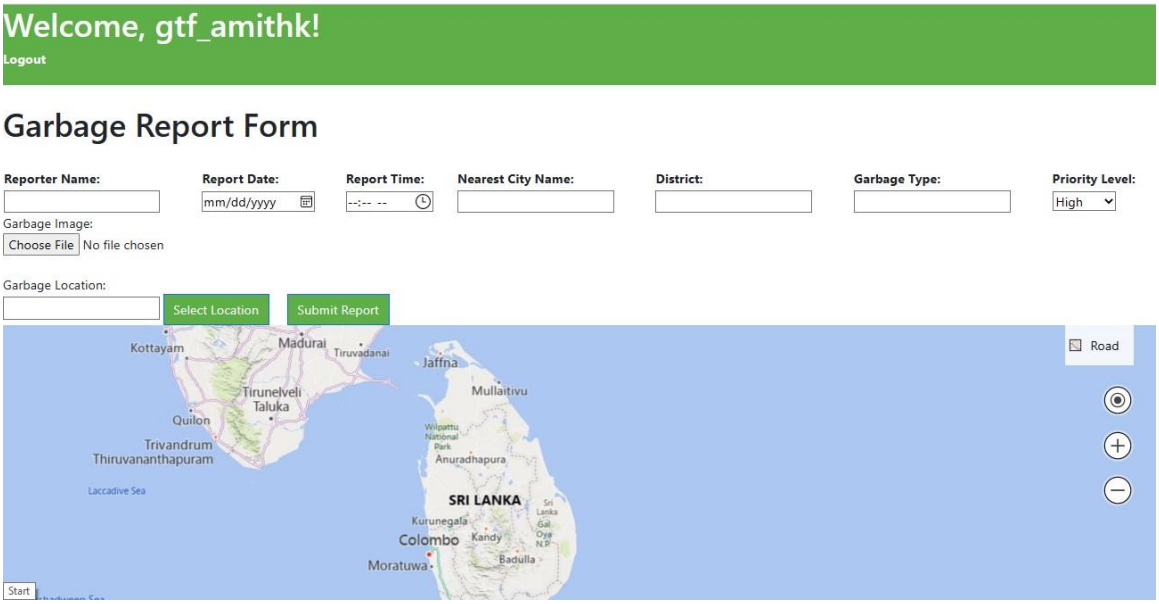


Figure 4 Data insert page of gtf member

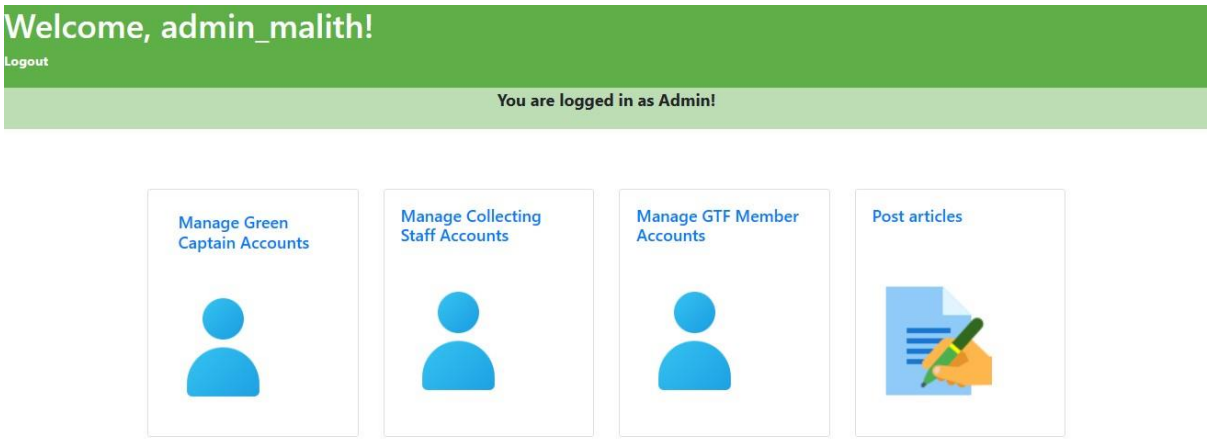
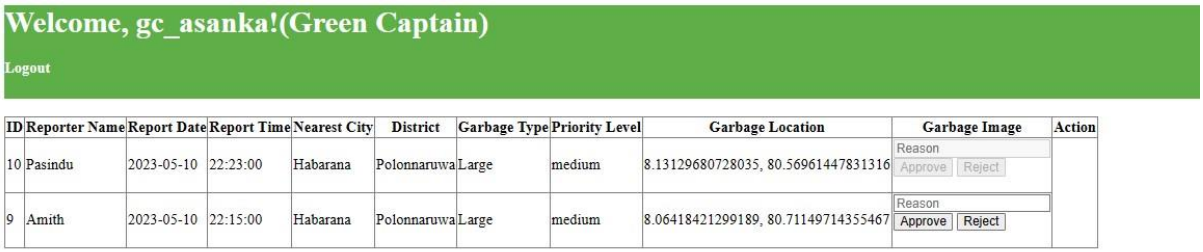


Figure 5 Admin dashboard



ID	Reporter Name	Report Date	Report Time	Nearest City	District	Garbage Type	Priority Level	Garbage Location	Garbage Image	Action
10	Pasindu	2023-05-10	22:23:00	Habarana	Polonnaruwa	Large	medium	8.13129680728035, 80.56961447831316	Reason Approve Reject	
9	Amith	2023-05-10	22:15:00	Habarana	Polonnaruwa	Large	medium	8.06418421299189, 80.71149714355467	Reason Approve Reject	

Figure 6 Green Captain dashboard

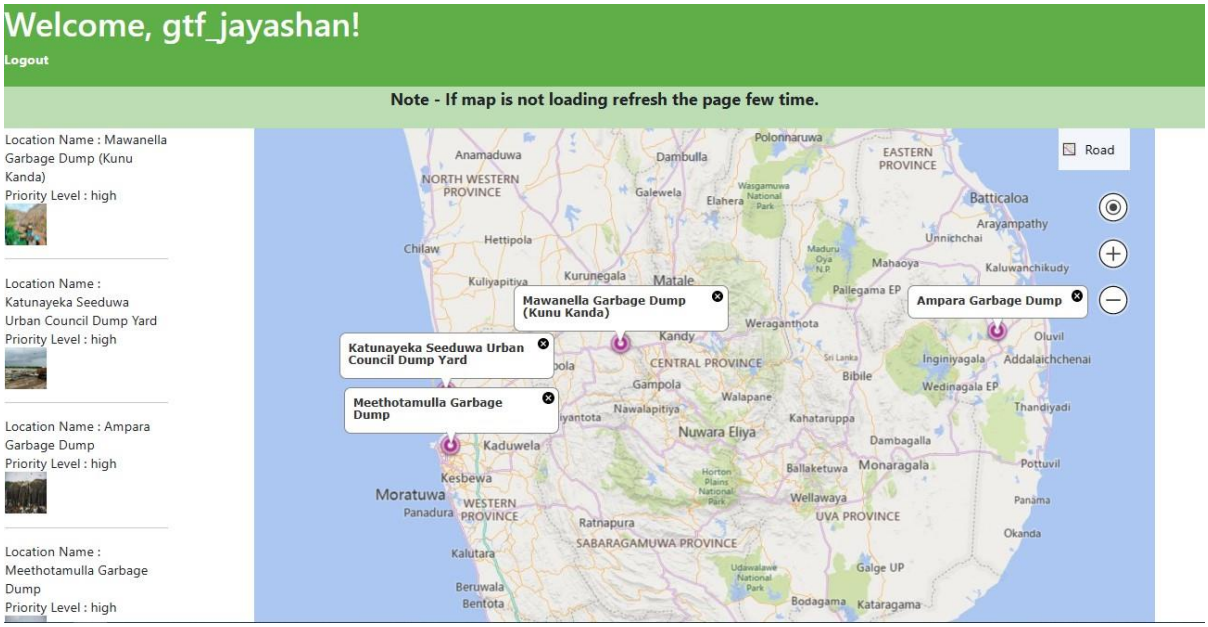


Figure 7 Collecting Staff member dashboard

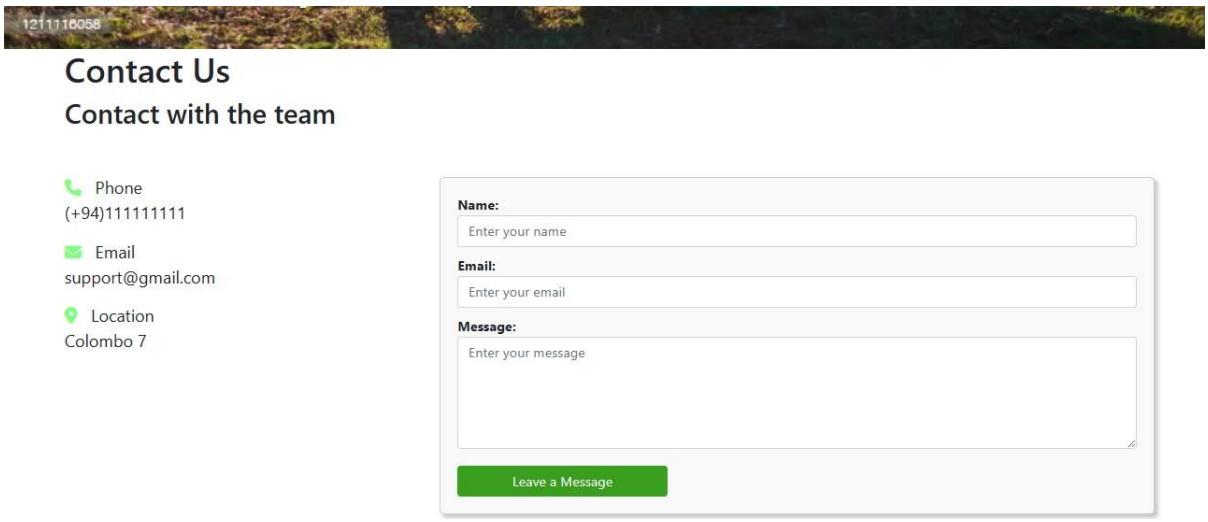


Figure 8 Contac Form

Contribution

Name - PKP Sandamal

Plymouth index - 10820873

Method of Contributed:

Frontend – I made contribution to build index.php page , repotertPage.php page ,Login.php page and Registration.php page. So here i did decoration by using CSS, JavaScript and bootstrap.. Also I combined all the php web page tother which are done by rest of group members.

Map – I made contribution for integrating map into our project .For this I used Map as Microsoft Bing map which can used free for educational purposes and here I used its API for integrating map into our website.

Backend – I made contribution to build user_login , user_registration,reportDetails and ReportInfo php pages. Here I integrated backend database as MYSQL db for our project.

Testing – Here I tested Login.php ,Registration.php and repoterPage.php pages individually. I insert all the incorrect and correct information for those pages and check whether actual result match with expected result. Further I did unit and integrated testing bases on Registration form.

Validation- I validate Login form and registration form. For this I used hash code for password which can used to inset password into database as encrypted one also, I used password pattern when user enter password into system through input fields which can be help for making strong password.

Test cases – I wrote test cases for Registration form validation ,User account creation,Article creation , Location marking php pages. Also I wrote test case for Registration form validation for doing unit testing and integration testing.

Name WHP Fernando

Plymouth index 10820893

Method of Contribution:

Frontend – I made contribution to build Admin ,About,Add-Articles,Contac php pages. So here i did decoration by using CSS, JavaScript and bootstrap

Backend – I made contribution to build save_article,contact_form, approve_Button php pages. Also connected each of pag mysql database successfully and also I made database and all the tables which are relevant to project

Testing – I tested save_article form which front end of Add_Artile php page by adding sample data.Also I tested contact form as I mentioned previously and also approve_Button php page.

Validation- I validate both Contac form. For this I used email pattern which need to verify whether email has standard format requirements and also add required attribute which can used to prevent empty filed

Test cases – I wrote test cases Login User Validation ,Registration User Validation ,Password Field Validation.Also I did mock test using python and check whether data is added into columns successfully.

Name EANP Alexander

Plymouth index 10818870

Method of Contribution:

Frontend – I made contribution to build Collecting_Staff and Terms php pages. Here i did decoration by using CSS, JavaScript and bootstrap.

Backend - I made contribution to build All_Article php page.

Test cases – I wrote test cases web page responsiveness.

Name PSM.Niranjana

Plymouth index 10820880

Method of Contribution:

Frontend – I made contribution to build FAQ and Videos php pages. Here i did decoration by using CSS, JavaScript and bootstrap.

Backend – I made contribution to build Report_Approvel php page.

Test cases – I wrote test cases for password field encryption.

Name FF Ahamad
Plymouth index 10818867

Method of Contribution:

Frontend – I made contribution to build Services php page. Here I did decoration by using CSS, JavaScript and bootstrap.

Backend - I made contribution to build Captain_Account php page.

Test cases – I wrote test cases for Report approval.