American International University-Bangladesh (AIUB)
Department of Computer Science
Faculty of Science &Technology (FST)

Summer 21 22

Section: E

Software Quality Assurance and Testing

BUSINESS SOLUTIONS

A Report submitted
By

SN	Student Name	Student ID
1	MD. Iktedar Hasan Rushdi Rajin	19-39757-1
2	Sk Shehrin Ehsan	19-39766-1
3	Srijon Barua	19-39744-1

Under the supervision of

Abhijit Bhowmik
Associate Professor
Department of Computer Science
Faculty of Science & Technology
American International University-Bangladesh

Software Test Plan for

BUSINESS SOLUTIONS

Version 1.0 approved

Prepared by

MD. Iktedar Hasan Rushdi Rajin

Sk Shehrin Ehsan

Srijon Barua

American International University-Bangladesh

20 August 2022

Checked by Industry Personnel

Name:
Designation:
Company:
Sign:
Date:

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2022.08.02	Sk Shehrin Ehsan	First Draft
0.2	2022.08.04	MD. Iktedar Hasan Rushdi Rajin	Second Draft
0.3	2022.08.08	Srijon Barua	Third Draft
0.4	2022.08.09	Srijon Barua	Fourth Draft
0.5	2022.08.12	MD. Iktedar Hasan Rushdi Rajin	Fifth Draft
0.6	2022.08.15	Sk Shehrin Ehsan	Sixth Draft
0.7	2022.08.16	Srjon Barua	Seventh Draft
0.8	2022.08.18	MD. Iktedar Hasan Rushdi Rajin	Eighth Draft
0.9	2022.08.19	Sk Shehrin Ehsan	Ninth Draft

1. TEST PLAN IDENTIFIER: RS-MTP01.3

2. REFERENCES

- Automated testing with Postman. https://www.postman.com/automated-testing/
- CRUD API resources by Surfside Media. https://www.youtube.com/watch?v=3Uy0KRPHQik&t=1127s
- Software quality and testing slides.

3. INTRODUCTION

Background to the Problem

Nowadays the social media platforms offer various features and functionalities. Companies and organizations utilize these features to the best of their abilities to advance their businesses. Many growing companies have used social media apps and websites to gain a good amount of exposure to jumpstart their businesses. However, because of the ever presence of businesses in today's social media platforms, the experience for many non-business users has become quite jarring. This is evidenced by the migration of over 1 million teenage users from Facebook to Snapchat and Tiktok. Twitter too, is facing this decline. These users prefer to use social media without advertisements and solely for communication. Facebook's Marketplace was developed to isolate the businesses from the main platform. However, Facebook and Instagram are still being used to further both small and large businesses.

Solution to the Problem

We propose a one-stop solution for businesses to connect with others: a website dedicated to build connections between businesses. Instead of paying a steep cost for marketing and advertisement on every social media platform, all information regarding the business will be available on the site. It will also have the option to view other businesses.

This particular software, if used as intended, would discourage businesses to advertisement spamming in various platforms. This will help users feel more comfortable within the social media platforms, which is one of the main goals of this software. It will also help businesses keep track of their social media progress much easily as data will not be spread in various sites.

LinkedIn is a similar existing solution to this problem. However, our solution differs in the fact that our software will be business centric while LinkedIn caters to individuals without businesses as well.

4. REQUIREMNT SPECIFICATION

4.1 System Features

1. Login
Functional Requirements

1.1 The software shall allow users to login with their given email and password
1.2 If the email and/or password has been inserted wrong user must not be able to login.

Priority Level: High
Precondition: user must have valid user email and password

2. Registration
Functional Requirements

2.1 New user must be able to insert name, email and password to register themselves.
2.2 If the name/ email/ password is empty then user must not be able to register.
2.3 If name is less than 10 characters or greater than 50 characters it should be invalid.
2.4 Email must have a valid format.

Priority Level: High

3. Dashboard
Functional requirements

3.1 After successful login user must be able to see their dashboard.

Priority Level: Medium
Precondition: user must be logged in.

4. Creating a post
Functional Requirements

4.1 Logged in user must be able to create post with a title, slug, description and an image.
4.2 If title, description, slug or image is empty then the post won't be created.

Priority Level: Medium
Precondition: user must be logged in.

5. Viewing all the posts
Functional Requirements

5.1 Logged in user must be able to view all the posts that are made by him or other authorized users.

Priority Level: Medium
Precondition: user must be logged in.

6. Commenting on a single post
Functional Requirements

6.1. Logged in user must be able to comment in a single post.

Priority Level: Medium
Precondition: user must be logged in.

7. Viewing own post
Functional Requirements

7.1 Logged in user must be able to view his own posts.
7.2 While viewing his posts he might also be able to edit or delete his own posts.

Priority Level: Medium
Precondition: user must be logged in.

8. Editing own post
Functional Requirements

8.1 Logged in user must be able to edit his created posts, change title, slug, description or even picture of a post.

Priority Level: High
Precondition: user must be logged in.

9. Deleting own post
Functional Requirements

9.1 Logged in user must be able to delete his own post.

Priority Level: Medium
Precondition: user must be logged in.

10. User settings
Functional Requirements

10.1 Logged in user must be able to view user settings where they will find their dashboard, option to delete and edit his account.

Priority Level: High
Precondition: user must be logged in.

11. Editing account information

Functional Requirements

11.1 User can edit his name, email and password and update is account.
11.2 Must be valid name and email.

Priority Level: High
Precondition: user must be logged in.

12. Deleting account
Functional Requirements

12.1 User must be able to delete his account
12.2 After deletion user will be redirected to login page

Priority Level: High
Precondition: user must be logged in.

13. Logout
Functional Requirements

13.1 Logged in user must be able to logout successfully.
13.2 All the sessions will be cleared after logout.

Priority Level: High
Precondition: user must be logged in.

4.2 System Quality Attributes

USABILITY:
All the features in the software are very easy to use. The interface is designed in a way that makes the software very easy to navigate and learn. A regular user of ours should be able to post about their products or businesses on an average of 1 minute and a maximum of 3 minutes. The software can also be easily used by new or any non-technical business personnel.

RELIABILITY AND CORRECTNESS:
As per our test results, our software does not produce incorrect results. The redirection of web-pages work accurately. The software does not crash randomly and does not freeze as it is lightweight. Therefore, our software is quite reliable.

MODULARITY:
We have maintained MVC (Model-View-Controller) architecture within our software. The modules are separated to keep the whole system fail-safe when one section faces a bug or error. Prioritizing modular architecture has allowed our testers and developers to easily detect and isolate bugs during testing phase.

In the future, if we wish to add any new feature, that will also be much easier due to modularity.

MAINTANABILITY:
Due to clean and organized code, modular architecture and proper documentation, our team has been able to easy maintain the system. Some of the maintenance tasks carried out by the maintenance team are: solving bugs and errors, changing or adding features. With just over 24 hours, our team shall be able to modify or correct any features.

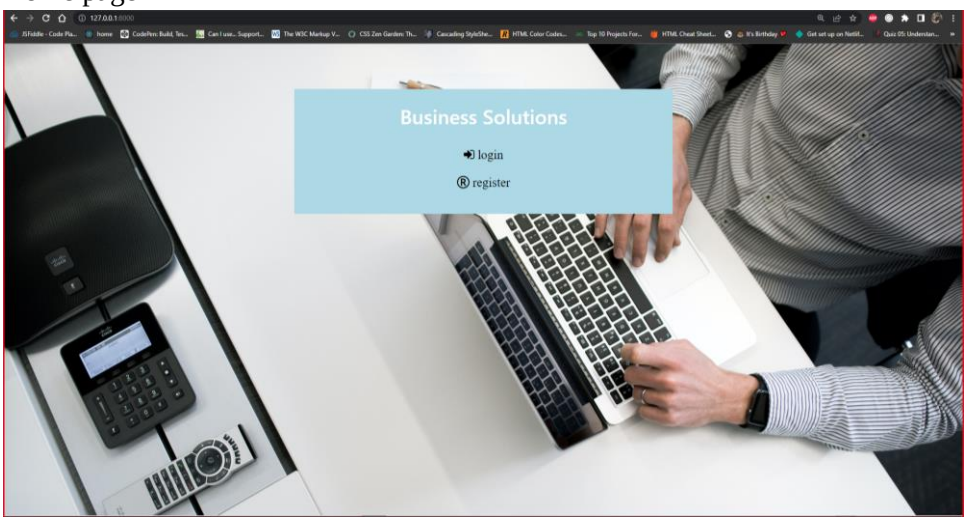
EFFICIENCY:
Our system has very high efficiency as it takes very little time to navigate it and to utilize it for its purposes. Being lightweight and fast has allowed it to be very efficient.

TESTABILITY:
Testability is heavily affected by modularity. Our system's modularity allows it to be very easily tested. Bugs and errors can easily be detected and isolated.

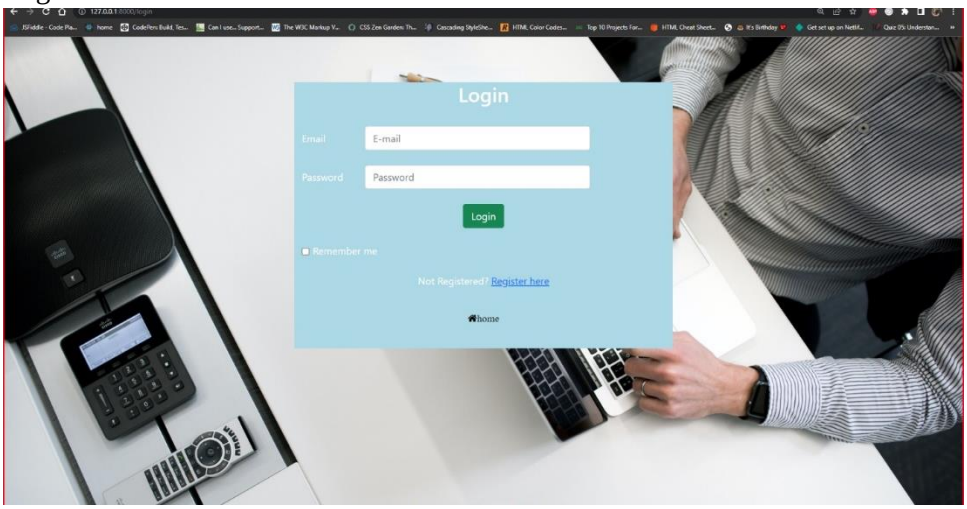
FLEXIBILITY:
A maintenance programmer who has at least six months of experience shall be able to produce modifications and fix bugs and also conduct testing, with no more than 2 hour of labor.

4.3 System Interface

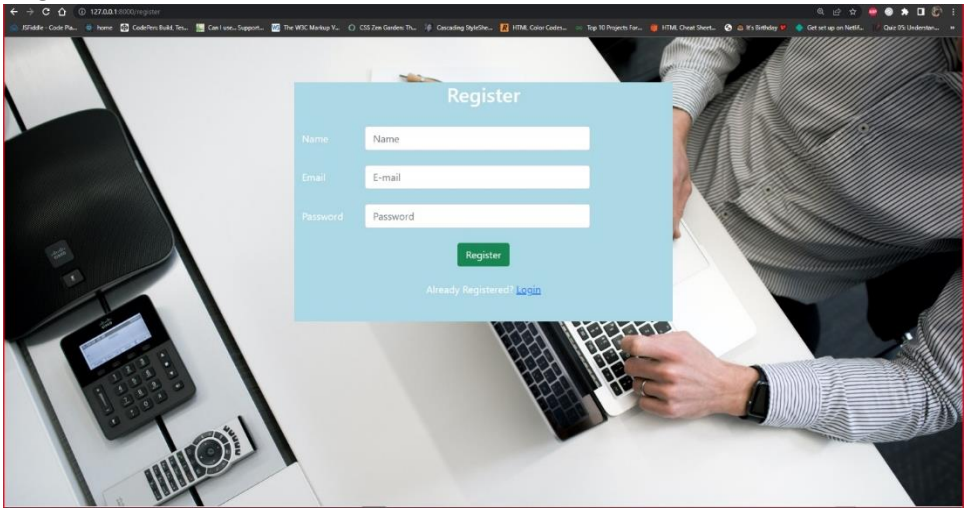
- Home page



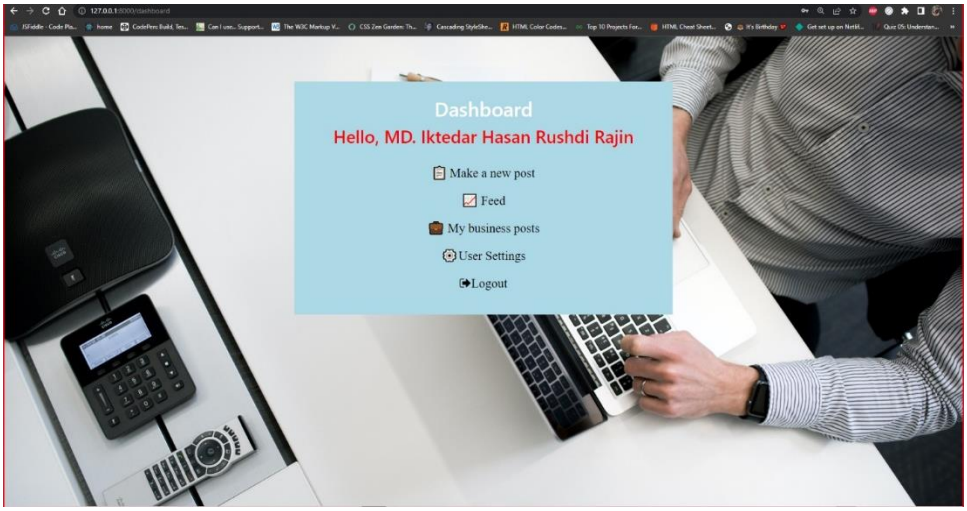
- Login



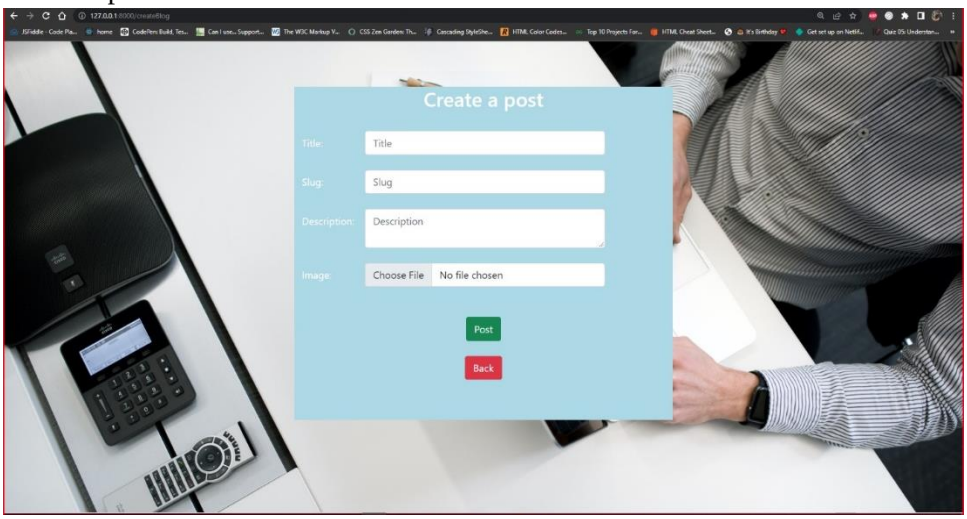
- Registration



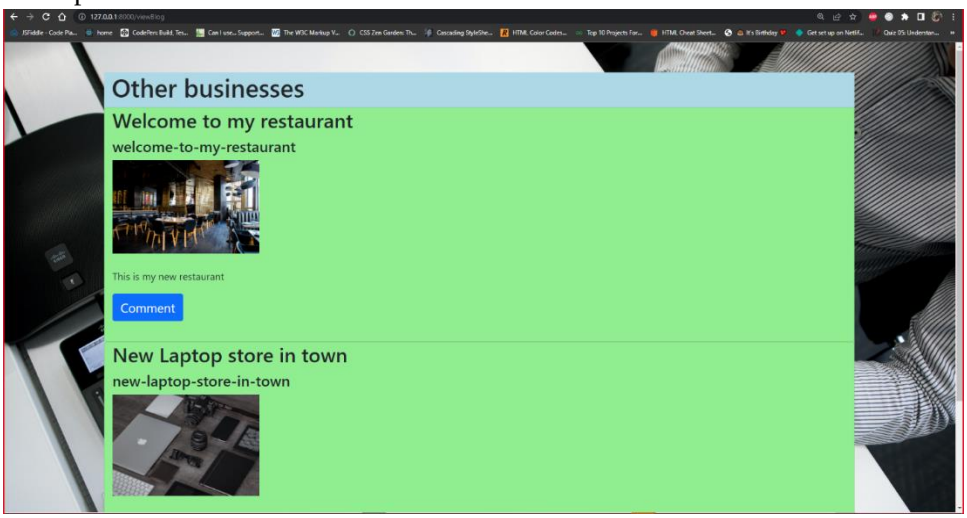
- Dashboard



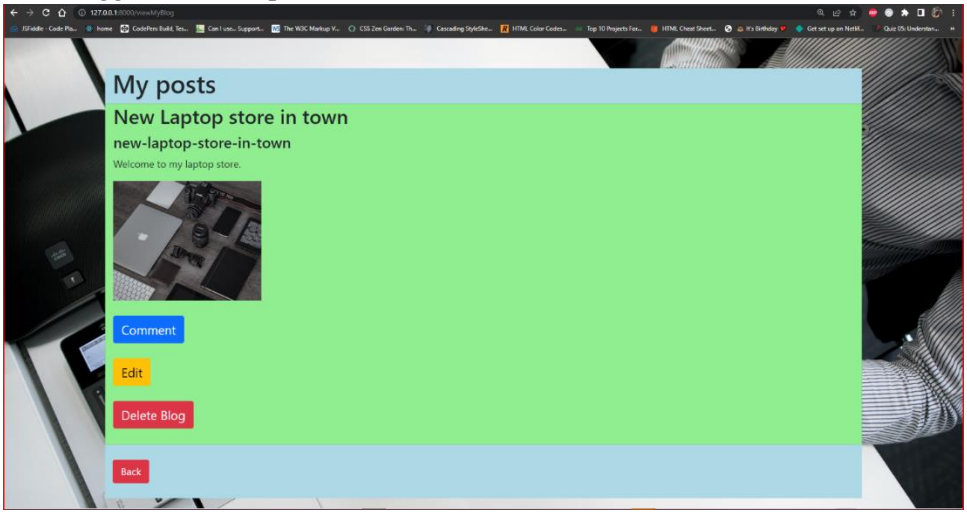
- Create a post



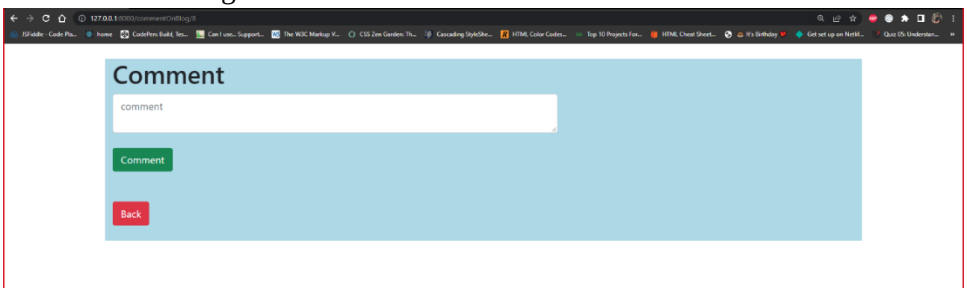
- View posts of all users'



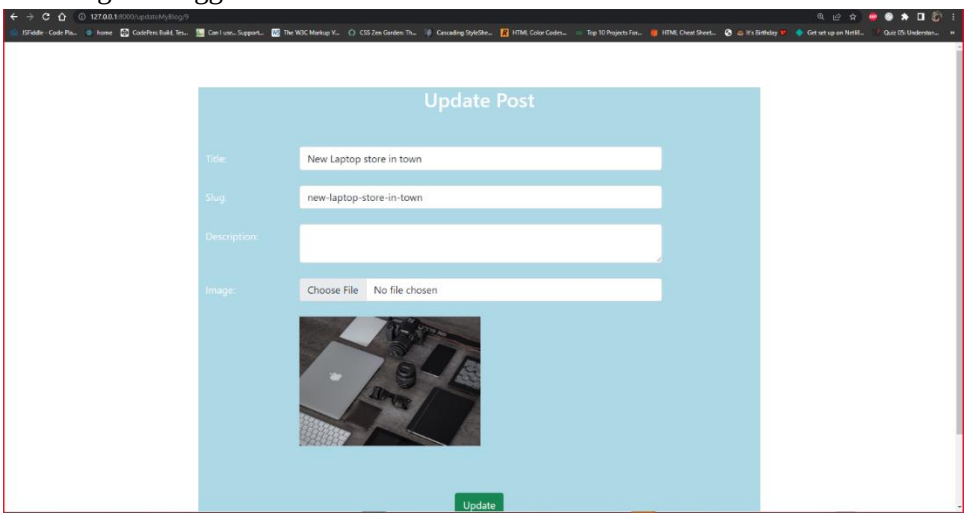
- View logged in user's post



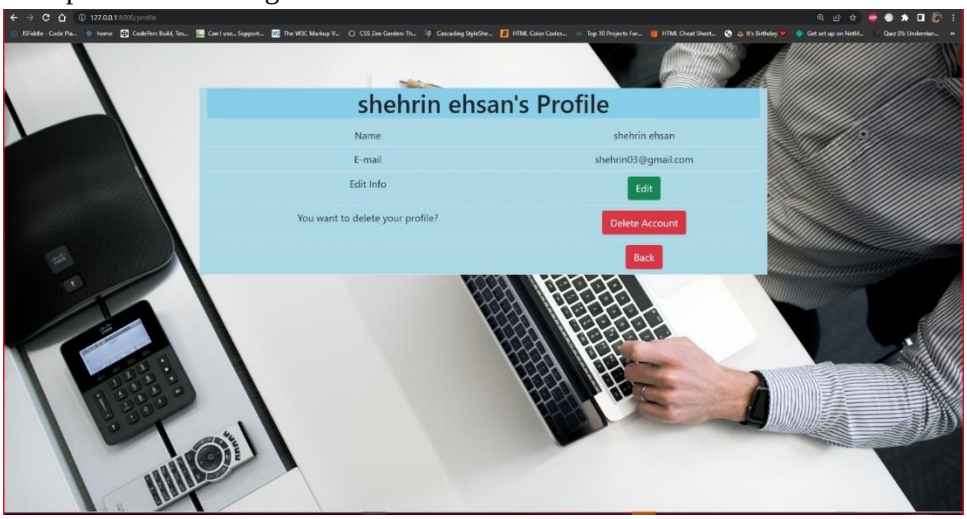
- Comment on a blog



- Edit blog of a logged in user



- User profile and settings



- Edit user's information

4.4 Project Requirements

Effort Estimation:

Our project is to develop an application named "Business Solutions".

Development Time = 1 Month

Required number of people = 3

Budget Estimation:

Duration in weeks = 4 weeks

Office days = 5 days

Working hours = 7 Hours

Working hour per week = (5*7) hours = 35 hours

Total Working hours = (35*4) hours = 140 hours.

Developer salary = 1000 Taka

Total developers' salary = (1000*140) Taka = 140000 Taka

Expenditure	Unit Amount	Total Amount
Salary for 4 developer		320000 Taka
3 months office rent	3*10000	30000 Taka
Electricity and other costs		20000 Taka
1 months Maintenance cost		15000 Taka

Travel Cost		5000 Taka
Total Cost		390000 Taka

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts. For example:

- **Viewing own posts:** We have a feature where logged in user can see his own posts. We did not test the feature already cause user can already see his posts in the feed.
- **Editing own posts:** There is a feature in our system where user can view his posts and edit the post. We haven't tested the feature in this release.
- **Deleting own posts:** Logged in users can also delete his posts. But we did not test the feature in this release.

Although we tested all the above-mentioned features with API but while editing or viewing or creation, we cannot upload the image properly due to the shortcomings in programming in back-end.

Also, there are some user interface issues that were seen during unit testing. "Editing user's information", "Editing user posts", "Commenting on a single post" has some issues with user interface.

6. TESTING APPROACH

6.1 Testing Levels

UNIT TESTING

This will be done by the software developer. Each developer will test each of the modules that they have created upon completion of that module. Sufficient documentation of unit testing has to be created and sent to Test Engineer. The results will be authorized by Development Team Lead. White box testing will be conducted.

INTEGRATION TESTING

Modules will be grouped and their integration will be tested. They will be tested for logic and connectivity; that is; the modules will be tested for working properly in association with other modules. The interaction of these modules will be tested. Technique will be Bottom Up testing.

SYSTEM TESTING

Testing will be performed by the Testing Manager and Development Manager in the presence of the software developers who were responsible for the project. The complete system will be tested; complete system integration will be verified. During this testing phase, critical defects will be found within the whole system. The technique will be black box testing.

ACCEPTANCE TESTING

This testing phase will be conducted by the help of end-users. It will be done to see how the market responds to the software and their feedback will be recorded. Any defects or bugs that

were not found during the previous phases of testing will be detected. The technique will be black box testing.

6.2 Test Tools

- **Postman**

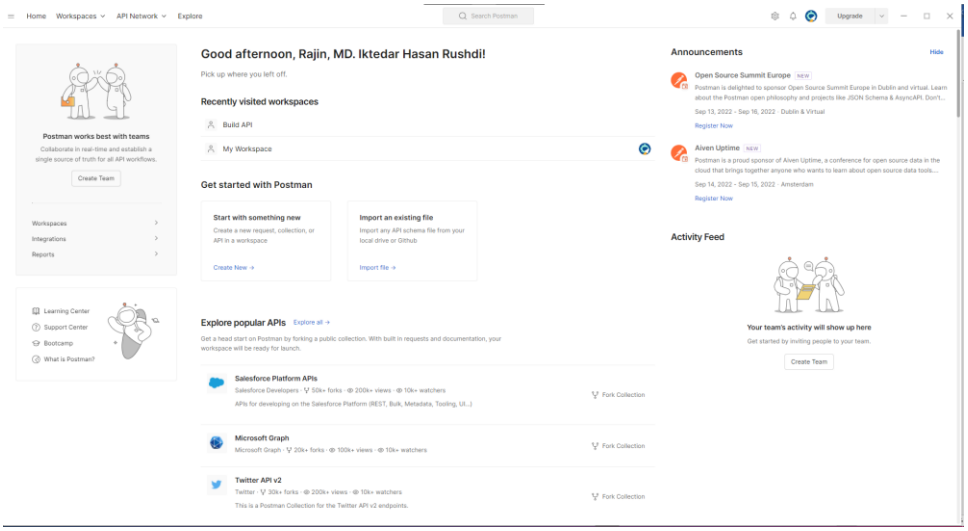


Figure 1: Postman homepage

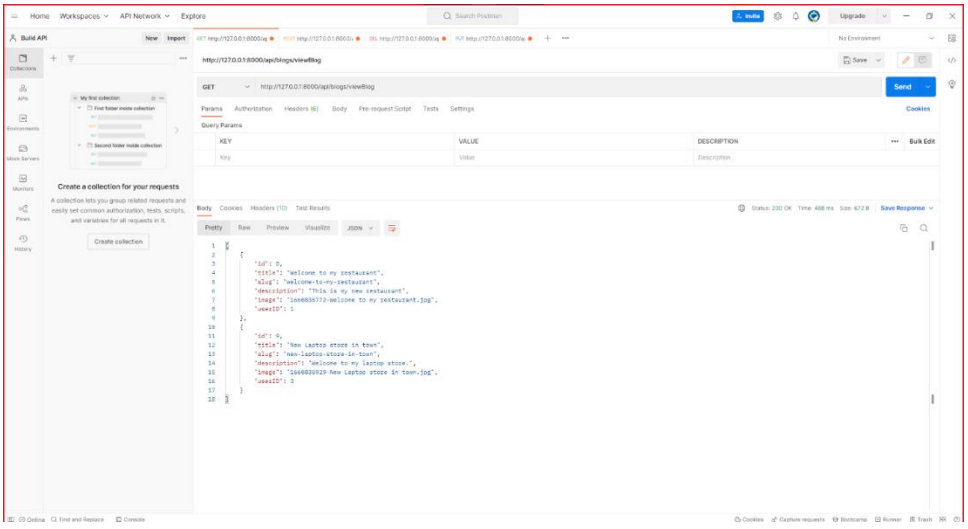


Figure 2: View all blogs

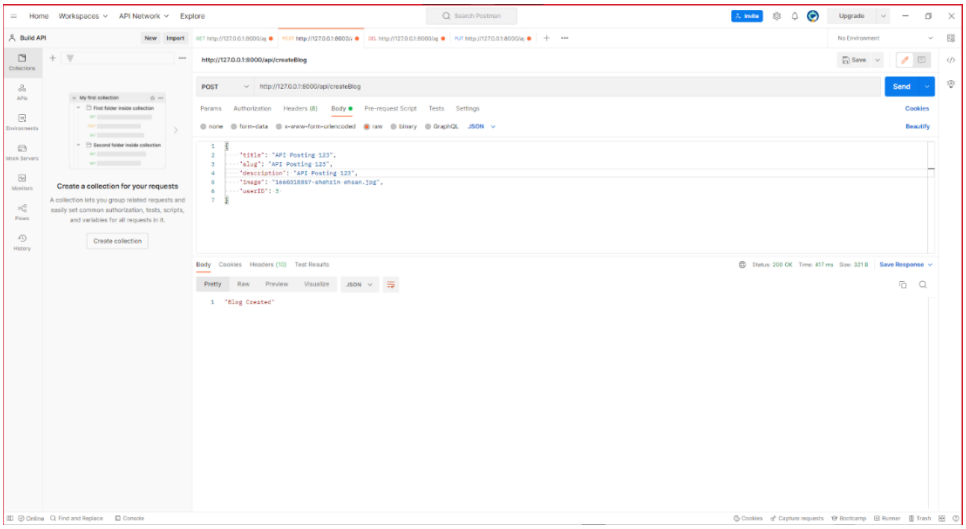


Figure 3: Create blog

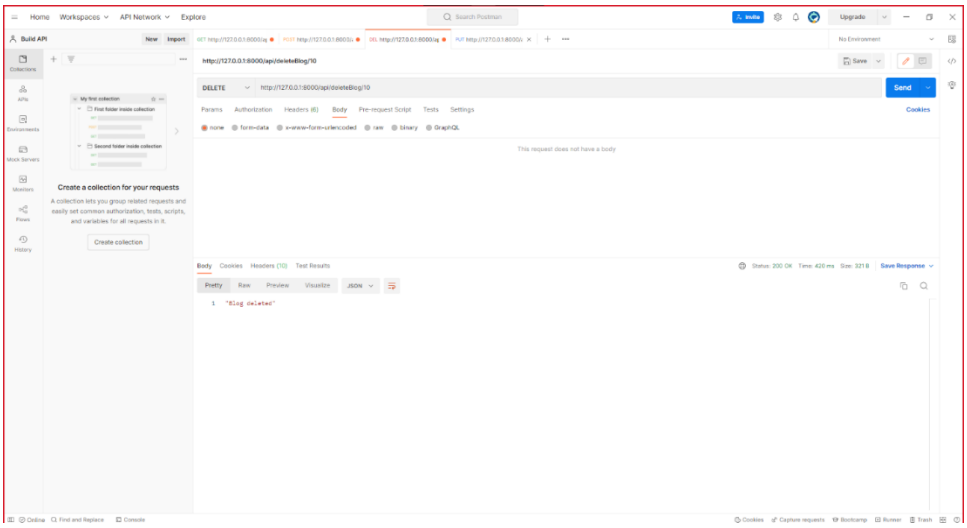


Figure 4: Delete a blog

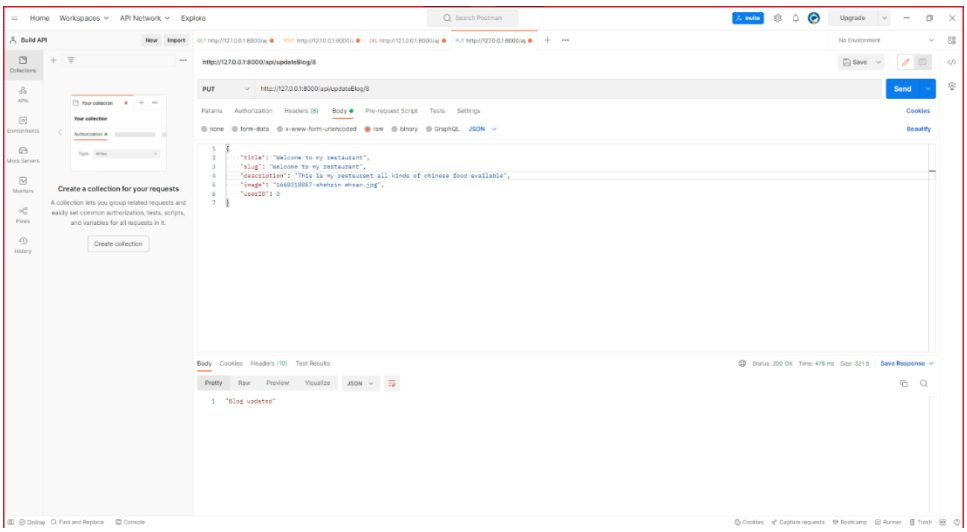
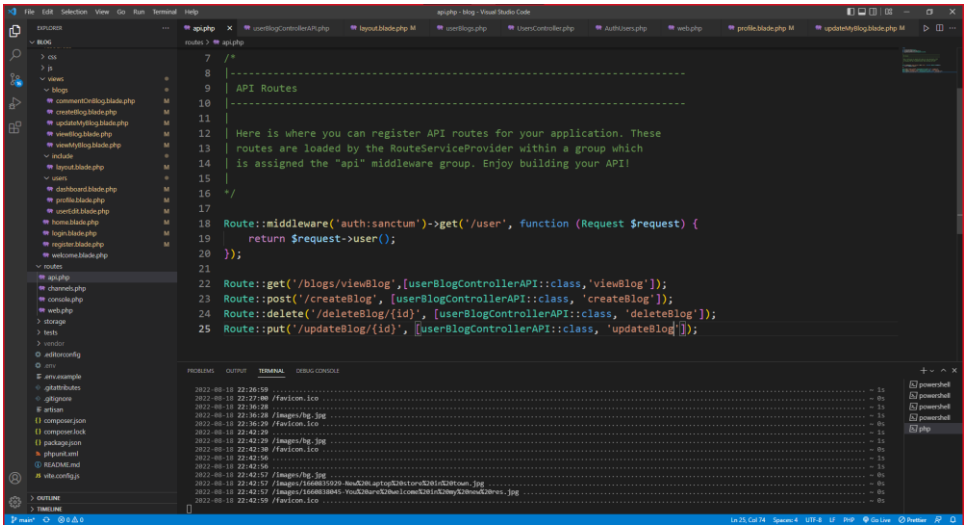


Figure 5: Update a blog



```
/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('/blogs/viewBlog',[userBlogControllerAPI::class,'viewBlog']);
Route::post('/createBlog', [userBlogControllerAPI::class, 'createBlog']);
Route::delete('/deleteBlog/{id}', [userBlogControllerAPI::class, 'deleteBlog']);
Route::put('/updateBlog/{id}', [userBlogControllerAPI::class, 'updateBlog']);
```

Figure 6: Routing for API

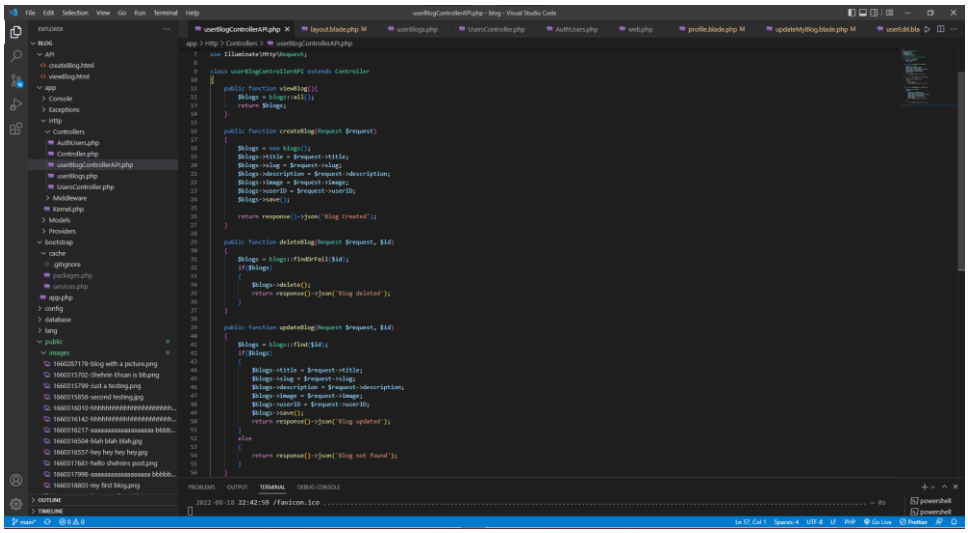


Figure 7: Code for API

6.3 Meetings

We believe that proper designation of team members and proper scheduling of tasks increases productivity and efficiency exponentially. In order to do this properly, team members must be evaluated and they must discuss their strengths and weaknesses openly. This will be done in a week-start meeting. Progress will be evaluated and any decisions about schedule and planning changes in the week-end meeting. These two meetings will include the entire project team. The test team will meet at the end of the week once a week to identify patterns and relations in the errors to prevent as much of bugs possible. The test team leader and the development team leader will meet with the project manager once every two weeks to discuss progress. Additional meetings may be called for emergencies.

7. TEST CASES/TEST ITEMS

Project Name: Business Solutions			Test Designed by: Srijon Barua	
Test Case ID: FR_2			Test Designed date:10/08/2022	
Test Priority (Low, Medium, High): Medium			Test Executed by: Srijon Barua	
Module Name: Registration Session			Test Execution date:09/08/2022	
Test Title: Registering an account in the website				
Description: Test the registration page				
Precondition (If any):				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Go to the website 1. Enter all the necessary user details 3.Click submit	Name: John Email: john@gmail.com Password: 1234	User will redirect to the login page	As expected,	Pass
Post Condition:				

Project Name: Business Solutions			Test Designed by: Srijon Barua	
Test Case ID: FR_1			Test Designed date:09/08/2022	
Test Priority (Low, Medium, High): High			Test Executed by: Shehrin Ehsan	
Module Name: Login Session			Test Execution date:10/08/22	
Test Title: verify login with valid username and password				
Description: Test website login page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Enter username 3. Enter password 4. Click submit	Username: rajin2410@gmail.com Password: 12345	User should login into the application	As expected,	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

Project Name: Business Solutions			Test Designed by: Srijon Barua	
Test Case ID: FR_13			Test Designed date:09/08/2022	
Test Priority (Low, Medium, High): High			Test Executed by: Srijon Barua	
Module Name: Log out Session			Test Execution date:	
Test Title: logging out from the website				
Description: Test log out option of the website.				
Precondition (If any): User must have to login to the website				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Select the logout option.		User should logged out from the application.	As expected,	Pass
Post Condition:				

Project Name: Business Solutions			Test Designed by: Sriion Barua	
Test Case ID: FR_5			Test Designed date:09/08/2022	
Test Priority (Low, Medium, High): Low			Test Executed by: MD. Iktedar Hasan Rushdi Rajin	
Module Name: Viewing all post			Test Execution date: 14/08/22	
Test Title: View all the post from the feed				
Description: Test the viewing all post option of the website.				
Precondition (If any): User must have to login to the website				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. User need to login to the website. 2. Select feed option		Will redirect to the feed page	As expected,	Pass
Post Condition:				

Project Name: Business Solutions	Test Designed by: MD. Iktedar Hasan Rushdi Rajin
Test Case ID: FR_6	Test Designed date: 10/ 08/22
Test Priority (Low, Medium, High): Low	Test Executed by: SK Shehrin Ehsan
Module Name: Commenting	Test Execution date: 15/ 08/22
Test Title: Comment on a single blog.	
Description: Logged in users can comment on other users' posts.	
Precondition (If any): User must be logged in.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in 2. From dashboard go to "Feed" 3. A page with all post will arrive 4. Under each post there is a comment button. Click on that. 5. A page with comment field will arrive. 6. User will be able to comment on that single post.	Good post.	Comment will be submitted.	As expected,	Pass
Post Condition: Comment will be submitted in the database.				

Project Name: Business Solutions	Test Designed by: MD. Iktedar Hasan Rushdi Rajin
Test Case ID: FR_12	Test Designed date: 10/ 08/22
Test Priority (Low, Medium, High): High	Test Executed by: MD. Iktedar Hasan Rushdi Rajin
Module Name: Deleting profile	Test Execution date: 13/ 08/22
Test Title: Delete logged in user's account.	
Description: Logged in users can delete their account.	
Precondition (If any): User must be logged in.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in 2. From dashboard go to "User settings" 3. Profile page will arrive 4. Click "Delete Account" button 5. User account will be deleted 6. User will be logged out redirected to login page	none	Account should be deleted.	As expected,	Pass

Post Condition: User's information will be deleted from the database. User will not be able to login with their previous information.

Project Name: Business Solutions	Test Designed by: MD. Iktedar Hasan Rushdi Rajin
Test Case ID: FR_11	Test Designed date: 10/ 08/22
Test Priority (Low, Medium, High): High	Test Executed by: SK Shehrin Ehsan
Module Name: Editing profile	Test Execution date: 14/ 08/22
Test Title: Editing profile details of logged in users	
Description: Logged in users can edit personal information.	
Precondition (If any): User must be logged in.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in 2. From dashboard go to "User settings" 3. Profile page will arrive 4. Click Edit button 5. There change Name, email or password 6. Click update.	name: MD. Iktedar Hasan	User's name should be updated.	As expected,	Pass
Post Condition: User's information will be updated in the database. User will be able to see the changed name in dashboard and profile.				

Project Name: Business Solutions			Test Designed by: Sk Shehrin Ehsan	
Test Case ID: FR_1			Test Designed date: 15/ 08/ 2022	
Test Priority (Low, Medium, High): Medium			Test Executed by: MD. Iktedar Hasan Rushdi Rajin	
Module Name: Login Session			Test Execution date: 18/ 08/ 2022	
Test Title: Verify post creation				
Description: Checking if creating a post works				
Precondition: User must have an account.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Log in 3. Click on "Make a new post" 4. Fill up information 5. Click "Post:	Title: Business Name Slug: business-name Description: This is my business Image: img.jpg	The page redirects to Dashboard and the post is visible in "My Business Posts"	As expected,	Pass
Post Condition: User is validated with database and successfully logged in to account. The account session details are logged in the database.				

8. ITEM PASS/FAIL CRITERIA

For our project we have implemented a total of 8 test cases. At first, when applied the test case to system 85% of the test cases were passed successfully and 15% were fail.

9. TEST DELIVERABLES

- Screen prototypes
- Test plan
- Test results documents
- Test summary
- Errors
- Bug report

10. STAFFING AND TRAINING NEEDS

PROJECT MANAGER:
Responsible for overseeing the entire project cycle; taking decisions to mediate concerns between development team and testing team.

TEST MANAGER:
Responsible for overseeing entire testing process; reporting progress to Project Manager; collaborate with Development Manager in order to create efficient schedules.

TEST ENGINEERS:
Responsible for planning tests, creating test methods, test information, running tests, examining episodes, writing mechanized test strategies.

SOFTWARE ENGINEERS:
Responsible for front end and back-end development of the software.

TRAINING

- Training will be required for Sales and Marketing teams on the usage of the software.
- Non-technical Administration personnel will require training in comprehending and writing reports.

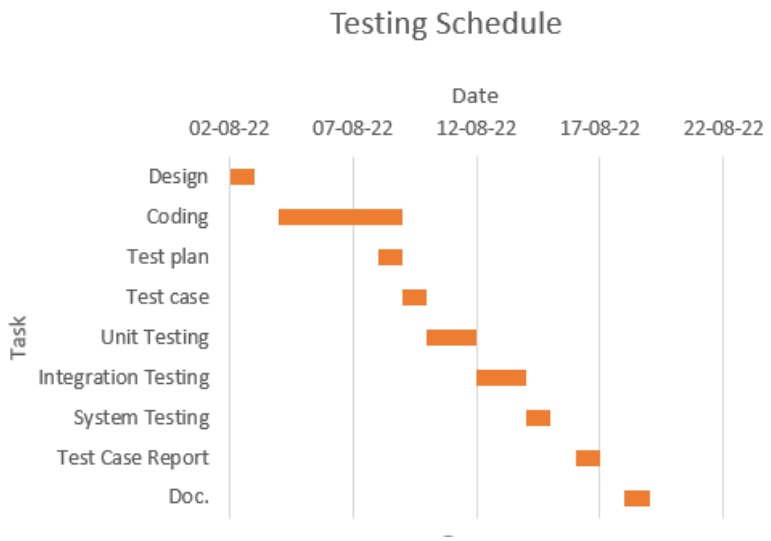
11. RESPONSIBILITIES

Name	***Role***	***Responsibilities***
MD. IKTEDAR HASAN RUSHDI RAJIN	Software Engineer	1.Designing and implementing software programs. 2. Writing and implementing efficient code. 3. Writing and executing test cases 4. Making documentation.
SK SHEHRIN EHSAN	Quality Analyst	1. Creation of test plans, test forms, test cases and test information. 2. Execution of test cases. 3. Execute all the test cases and report defects. 4.Prepare all reports and documentation
SRIJON BARUA	Test Engineer	1. Prepare the report of testing activities. 2. Develop test cases and prioritize testing activities. 3. To check in the event that the group has all the fundamental assets to execute the testing exercises.

12. TESTING SCHEDULE

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan timeline. The persons required for each process are detailed in the project timeline and plan as well. Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the project manager in conjunction with the development and test team leaders. Schedule must be done using any PM tool.

Task name	**Duration**	**Responsible**
Design	1	Project Manager
Coding	5	Software developer
Test plan	1	Testing team
Test case	1	Quality analyst
Unit Testing	2	Software developer
Integration Testing	2	Testing team and development team
System testing	1	Testing team
Test case report	1	Test Engineer
Documentation	1	Testing Engineer



13. PLANNING RISKS AND CONTINGENCIES

- Software development cycle may take longer than planned. This may set back testing schedule.
- Testing Team staff shortage may mean that bugs may go unnoticed until Acceptance Testing.
- Integration testing may take longer than expected.

14. APROVALS

Project Sponsor	MD. Iktedar Hasan Rushdi Rajin
Development Management	MD. Iktedar Hasan Rushdi Rajin
EDI Project Manager	Sk Shehrin Ehsan
RS Test Manager	Srijon Barua
RS Development Team Manager	MD. Iktedar Hasan Rushdi Rajin
Reassigned Sales	Sk Shehrin Ehsan
Order Entry EDI Team Manager	Srijon Barua