
Test Document

for

DigiSchool

Version 2.1

Prepared by

Group #: 19

Group Name: Dementors

GUGULOTH VARUN	180272	varunguguloth@gmail.com
SUMIT NAGLE	170732	sumit.nagle@gmail.com
GAURAV KUMAR	180265	phogat102135@gmail.com
MOTA JITENDRA	180434	jitendrathierider23@gmail.com
AKASH DAYANAND CHAVAN	180054	akashchavan180054@gmail.com
DIKSHANT RAJ MEENA	180247	dkrajmeena27899@gmail.com
SHREYASI PRASAD	190824	student.shreyasi@gmail.com
YASHWANT TAILOR	180894	yasht88101@gmail.com
YEDDULA VINAY CHANDU	191181	chandu8042439@gmail.com
AMAN RAW	190108	amanraw4444@gmail.com

Course: CS253

Mentor TA: Shatroopa Saxena

Date: 30-01-2022

CONTENTS	II
REVISIONS	II
1 INTRODUCTION	1
2 UNIT TESTING	2
3 INTEGRATION TESTING	3
4 SYSTEM TESTING	4
5 CONCLUSION	5
APPENDIX A - GROUP LOG	6

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is	00/00/00

Version	Primary Author(s)	Description of Version	Date Completed
		touched, only when the version is being upgraded.	

1 Introduction

Manual Testing was done. Which involved going through all the page, and trying to find bugs in it. It involved going through all the web paths involved in our web application. We also tested the database for data type validation so that garbage data is not avoided.

Testing was conducted in parallel with the implementation as well as few conducted after the implementation was completed. Thus allowed us to solve the bugs as soon as possible so that, nested bugs are eliminated/minimised.

Tester were the developers (i.e we ourself tested the web application), where we all brought our own knowledge of finding bugs. We also used some penetration techniques in order to find security breach.

Coverage criteria includes, **code coverage**, **statement coverage**, **Decision coverage** and **Branch coverage**. Where the majorly used coverages were decision and statement coverage.

2 Unit Testing

1. [testapp/views.py](#)

Unit Details: It includes all the particular functions, which serves the purposes of our [test functional requirement](#), it includes the views files, testPage, testUploaded, eachTestView, editTestPage, editTestUpload, answerUpload, answerCheckPage, scoreUpload.

Test Owner: Sumit Nagle

Test Date: [20/03/2022] - [21/03/2022]

Test Results:

- Functions found with minor syntax errors and wrong conditional statements errors.
- Handled the error of file opening and closing errors where opening of file (as “write”) causes data clean out of that file.

Structural Coverage: **Statement** coverage, **Decision** coverage.

Additional Comments: Only the front-end data handling is to be looked out. such as correct test title and description which satisfy our data validation checks.

2. [lectureapp/views.py](#)

Unit Details: It includes all the particular functions, which serves the purposes of our [lecture functional requirement](#), it includes the views files, lecturePage, createPage, lectureUploaded, eachLectures, editLecturePage, editLectureUpload

Test Owner: Guguloth Varun

Test Date: [20/03/2022] - [21/03/2022]

Test Results:

- Found out that every condition in a decision in the program has not taken all possible outcomes at least once.
- Handled the error which caused problems in the editing of the uploaded lecture page.
- All the conditions were check and all the missing conditions and decisions were added.

Structural Coverage: **Statement** coverage, **Decision** coverage, **Branch** coverage

Additional Comments: Checked all the cases where we get an error while updating and retrieving the data.

3. [forumapp/views.py](#)

Unit Details: It includes all the particular functions, which serves the purposes of our [forum functional requirement](#), it includes the views files, forumPage, createPage, forumUploaded, eachForumView, forumAnswerUpload

Test Owner: Guguloth Varun

Test Date: [23/03/2022] - [24/03/2022]

Test Results:

- Functions found with minor syntax errors and wrong conditional statements errors.
- Error in data handling of json file was cleared, and file storage of json files were optimised and orderly saved.

Structural Coverage: **Statement** coverage, **Branch** coverage, **Decision** coverage.

Additional Comments: All the errors from backends are cleared, and only the front-end data handling is to be looked out.

4. [courseapp/views.py](#)

Unit Details: It includes all the particular functions, which serves the purposes of our [announcement](#) functional requirement, it includes the views files, newsPage, createPage, newsUploaded, editnewsPage, editnewsUpload

Test Owner: Shreyasi Prasad

Test Date: [23/03/2022] - [24/03/2022]

Test Results: Found out that every condition in a decision in the program has not taken all possible outcomes at least once, the news page is directed to the forum page after the editing done by the teacher.

Structural Coverage: **Statement** coverage, **Decision** coverage, **Branch** coverage

Additional Comments: Getting error while teacher saving the edited version of the news. this issue is resolved now

5. [loginapp/views.py](#)

Unit Details: It includes all the particular functions, which serves the purposes of our [login](#) functional requirement, it includes the views files, homePage, signUpPage, signUpPosted, signUpOTPVerified, contactPage, contactPageSubmitted, loginPage, loginPageCheck, logoutPage, forgotPasswordPage, forgotPasswordSendEmail, forgotPasswordOTPVerify, changePassword

Test Owner: Sumit Nagle

Test Date: [23/03/2022] - [24/03/2022]

Test Results:

- Found out that we are getting error in the forgotPasswordPage and related functions like forgotPasswordSendEmail, forgotPasswordOTPVerify, changePassword. Which were not working properly.
- Error of sending and verifying OTP. But now it is resolved.
- Login page was vulnerable to attempted login attack, where already a login person can login as another user at the same time from same machine. Now it is resolved.

Structural Coverage: **Statement** coverage, **Decision** coverage

Additional Comments: Still getting error in the forgotPasswordPage and related functions like forgotPasswordSendEmail, forgotPasswordOTPVerify, changePassword.

6. [profileapp/views.py](#)

Unit Details: It includes all the particular functions, which serves the purposes of our [profile](#) functional requirement, it includes the views files, profilePage, editProfilePage, editProfilePagePosted

Test Owner: Mota Jitendra

Test Date: [23/03/2022] - [24/03/2022]

Test Results: All the lines in code are not used by the app, found out that every condition in a decision in the program has not taken all possible outcomes at least once.

Structural Coverage: Statement coverage, Decision coverage

Additional Comments: All the required functions are working fine.

3 Integration Testing

1. ./digischool

Module Details: After all the unit testing was done, we check if all the urls are properly mapping for our already tested view functions (across all apps), for that we used the files, urls.py settings.py

Test Owner: Sumit Nagle

Test Date: [23/03/2022] - [24/03/2022]

Test Results:

- All the url paths as well as the relations with other urls were checked out.
- settings configurations were checked out accordingly in assistance with the directory folder involved and the file names used.
- All the import statements were check.

Additional Comments: views.py were already tested in unit testing phase.

2. loginapp,course app, forum app,lecture app, profile app

Module Details: All the models file were checked, so that all tables and their schema are correctly configured. So that there is no database errors involved. It includes the models files,

login app consists of following classes, USER_SIGNUP_DATABASE, OTP_DATABASE, QUERY_DATABASE, TEACHER_CODE_MAPPING

course app consists of following classes, CLASS_COURSES_MAPPING, AVAILABLE_COURSES, ALL_ANNOUNCEMENT

forum app consists of following classes, ALL_FORUMS

lecture app consists of following classes, ALL_LECTURES

test app consists of following classes, ALL_TESTS

profile app consists of following classes, USER_PROFILE_DATABASE

Test Owner: Guguloth Varun, Sumit Nagle

Test Date: [23/03/2022] - [24/03/2022]

Test Results:

All the tables generated in the database by these models are working fine and are mapping each other successfully through foreignkeys, And all the above mentioned modules are connected to main module(digiapp) smoothly by integrating them in digischool/settings.py file.

Additional Comments: More information can collected by adding the other fields but we might do these additions in the future.

4 System Testing

1. Requirement:

2. **Processor:** 1. GHz +
3. **minimum RAM:** 128 MB
4. **OS:** Windows XP, Vista, 7+, Mac OS X 10.5+, or Ubuntu 10+
5. **recommended Screen resolution :** 1280x1024 or larger Application window size : 1024x680 or larger
6. **Internet connection :** Required

Test Owner: Sumit Nagle, Guguloth Varun

Test Date: [26/03/2022] - [27/03/2022]

Test Results: ping test were performed (in the development server as well as on deployed website).

All the browsers were tested, and our website was working on browsers like chrome, mozilla, edge, brave in all the operating systems mentioned above and minimum required specifications.

Time test were done on all the view functions. It was found that most of them were quite efficient and were running in milli seconds. And the database querying was also fast (due to the django's fast and efficient ORM).

There were some view functions which not much efficient as compared to other view functions, mostly where large files will uploaded or downloaded.

5 Conclusion

How Effective and exhaustive was the testing?

The unit testing was efficient and quite effective, as we followed similar path of calling view functions in all the app, which means error resolution in one app, made it easy to find error in other apps.

To conclude, we are quite satisfied with our testing, however we still look forward for other tester to test our web application, and find errors that we couldnt.

Which components have not been tested adequately?

The Announcement part and password update part is not adequately tested in the current project.

What difficulties have you faced during testing?

Making sure of the exhaustiveness of the testing took a lot of effort and were never satisfied, and occasionally tested for errors. And this was time consuming and required a lot of efforts.

How could the testing process be improved?

Testing process can be improved by automating the many tests instead of the manual testing.

Appendix A - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>