

Team Icarus Milestone 5

Who: *Benjamin Zaeske, Quinlin McNatt, Xuejuan Feng, Michael Wegner, Aidan Duggan, Tianfang Wang*

Title: Icarus Project

Automated Tests:

We will be using Phpunit to do automated unit tests on our code. Phpunit is an application (very similar to unit testing frameworks in other programming languages) that makes unit testing php code very easy and streamlined. We will use these unit tests in order to see whether or not our code properly updates things like posts, username and login validation, and small features such as hitting a like button to increment the number of likes on a particular post. These automated unit tests (like all unit tests) will help us debug minor errors in our code and make sure that our features are all performing as expected. The separate user acceptance tests (documented later in this pdf) will be used to test our web application on a higher level to make sure that features work well together and that our application runs smoothly.

To run our automated tests you must have phpunit installed on your machine. A link to a website that tells you how to do this is [here](#). Once phpunit is installed you have to then navigate to the application directory (in our github that is the folder called 'Application') and run: 'phpunit <unit test name here>' in terminal. This will run the desired automated unit test of choice.

Note that we have yet to fully implement all of our tests yet. However, our framework for automating tests is set up and does run, which will make it easy for us to add tests as we continue to code. The following is a screenshot of our automated unit tests running (next page):

```
Application — -bash — 80x24
[19:35:51 ~: cd Dropbox/SoftwareDevelopment/Icarus_Project/Application/
[19:36:30 ~/Dropbox/SoftwareDevelopment/Icarus_Project/Application: ls
ProfilePics          login_header.php      signup_student.php
addbio.php           send_bio.php          signup_teacher.php
connect.php          signin.js             style.css
footer.php           signin.php            testConnect.php
header.php           signout.php           user_homepage.php
index.php            signup.php
[19:36:32 ~/Dropbox/SoftwareDevelopment/Icarus_Project/Application: phpunit testC]
onnect.php
PHPUnit 7.1.4 by Sebastian Bergmann and contributors.

R                                                                    1 / 1 (100%)

Time: 62 ms, Memory: 8.00MB

There was 1 risky test:

1) ConnectTest::testConnection
This test did not perform any assertions

OK, but incomplete, skipped, or risky tests!
Tests: 1, Assertions: 0, Risky: 1.
[19:36:38 ~/Dropbox/SoftwareDevelopment/Icarus_Project/Application: ]
```

User Acceptance Tests:

1. User Acceptance test #1:

- **User story:** Create an account as a student/teacher and log in.
- **Prerequisites:** Valid connection to the database so that the created user can be added to it.

Step #	Test step/ input	Expected results
1.	Click 'sign up as student' or 'sign up as teacher' link	Able to click on sign up link. Directs you to correct signup page. (student or teacher depending on which type of account is being created)
2.	Enter test username	Able to type in username box
3.	Enter test password	Able to type in password box as well as the confirm password box.
4.	Enter email	Able to type in enter email box
5.	Click 'create account' button	Able to click the create account button and sends you to account created confirmation page.

6.	Login as the new user	Able to log in as the new user (authentication validates the created user as being in the database and takes you to the classroom list page after). Account type (student or teacher) is correctly displayed in the top right.
----	-----------------------	--

2. User Acceptance test #2:

- **User story:** Login as a student, post a question, and like a question.
- **Prerequisites:** Database has been filled with test users for login purposes as well as a test classroom so that the student can post.

Step #	Test step/ input	Expected results
1.	Login as one of the test accounts already in the database	Login authenticates user properly and brings them to classrooms list page
2.	Click on one of the test classrooms (already in the database)	Able to click on classroom and brings you to the correct forum/chatroom page.
3.	Click on chat box	Able to click on and type in chat box
4.	Post the question	Able to click on the post question button. Question is displayed in the chatroom. Text box is cleared.
5.	Like the question	Able to click the like button on the question and increment the number of likes on that particular question by 1.

3. User Acceptance test #3:

- **User story:** Log in as a teacher, create a new classroom and add students to the classroom.
- **Prerequisites:** Database has been filled with test data (a test teacher account) so that the teacher can be authenticated and perform the required tests.

Step #	Test step/ input	Expected results
1.	Log in as test teacher account	Able to login (database authenticates the test teacher) and brings you to the classrooms list page.

2.	Click on the create new classroom button	Teacher has access to the create new classroom button, and can click on it. Clicking on it brings them to the create new classroom page.
3.	Enter class name in text field.	Able to type in the text box.
4.	Click the 'create class' button.	Able to click the 'create class' button which brings the teacher back to the classrooms page (with the new classroom now visible)
5.	Click the enroll student button (located next to a particular class in the class list when logged on as a teacher)	Able to see the 'enroll student' button and can click on it. Brings you to the correct enroll student form.
6.	Enter student Id in text field	Able to type in the text field
7.	Click 'add student' button	Able to click the add student button. Database is correctly updated to show the new student as being in the class. Takes user back to the classrooms list page.