

Event Tracking System



Web based event tracking built
for organizers and volunteers.

TABLE OF CONTENTS

INSTALLATION.....	3
WAMP/XAMPP	
VERSION NUMBERS	
FEATURES	4
IMPLEMENTED FEATURES	
<i>Slots</i>	
<i>Events</i>	
<i>Organizations</i>	
MISSING FEATURES	4-5
KNOWN ISSUES.....	5
FUNCTIONALITY.....	5
BACKEND	
<i>Slot</i>	
<i>Event</i>	
<i>Organization</i>	6
<i>User</i>	
FRONTEND.....	
<i>Event Read</i>	6-7
<i>Event Create</i>	8
<i>Event Update</i>	9
<i>Event Delete</i>	10
TESTING COVERAGE.....	11
PROGRAM	#
READ TEST	12-13
CREATE/DELETE TEST	14-15
TEAM MEMBERS.....	16
TEAM INTRODUCTION.....	#
AMMAR HUSEINPAHIC	17
YIET MAI.....	18
NATANAEL V NISTOR.....	19
SURVEY	20
LICENSING AGREEMENT	21
SIGNATURES	22

Installation

WAMPServer and XAMPP are windows web development environments. They allow local hosting of web applications with apache, PHP and the MySQL database. They also come with phpMyAdmin and sqlitemanager for database management.

INSTALLING WAMP

DOWNLOAD LINK: <http://www.wampserver.com/dl.php>

1. Double click on the downloaded file and just follow the instructions. Everything is automatic. The WAMPServer package is delivered with the latest releases of apache, MySQL and php.
2. Once WAMPServer is installed, you can manually add additional components for apache, php or MySQL (only vc9, vc10 and vc11 compiled) versions.

USING WAMP

1. Run WAMPServer
2. The “www” directory will be automatically created (usually c:\wamp\www)
3. Create a subdirectory in “www” and put your project files inside.
4. Click on the “localhost” link in the WAMPServer menu or open your internet browser and go to the url : <http://localhost>

INSTALLING XAMPP

DOWNLOAD LINK: <https://www.apachefriends.org/index.html>

1. Double click on the downloaded file and just follow the instructions. You have the choice of installing various components such as MySQL, phpMyAdmin, Apache, and more from within the installer (Leave to default for this project).

USING XAMPP

2. The XAMPP Control Panel includes three main sections. In **Modules**, you will find all the services available. You can run each service by clicking the **Start** button.
3. The default settings should work for most people using XAMPP to create a testing environment to run a website for personal use or a client. However, depending on your setup configuration, you may need to change the TCP/IP port number for the Apache server, the database upload size, or set the password for phpMyAdmin. Click the **Config** button for those settings.

Version Numbers

The project was created on these version numbers. If you have errors in the future, try to roll back to one of these versions.

1. Apache – 2.4.27
2. PHP – 7.1.9
3. phpMyAdmin – 4.7.4
4. MySQL – 5.7.19

Features

IMPLEMENTED FEATURES

1. Backend
 - a. Slots
 - i. Create, Read, Update, Delete
 - ii. Paging
 - iii. Search Slot by ID
 - iv. Search Slot by Event
 - b. Events
 - i. Create, Read, Update, Delete
 - ii. Assign event to organization
 - iii. Assign slots to event
 - iv. Return slots belonging to event
 - v. Search Event by ID
 - c. Organizations
 - i. Create, Read, Update, Delete
 - ii. Register user to organization
 - iii. Return Events by Organization
 - d. Users
 - i. Create, Read, Update, Delete
 - ii. Log in with token validation
 - iii. Token expiration
2. Frontend
 - a. Example Documents
 - i. Reading items using API
 - ii. Using paging to read Slots from API
 - iii. Deleting a Slot using API
 - iv. Creating a Slot using API
 - b. AJAX/JQuery live table
 - i. Creating Events
 - ii. Viewing Events
 - iii. Updating Events
 - iv. Deleting Events
 - c. Login Page for users
 - i. Validates user info
 - ii. Assigns token to user
 - iii. Token checked

MISSING FEATURES

1. Backend
 - a. Token Validation for Creating, Updating, and Deleting
 - i. Slots
 - ii. Events
 - iii. Organizations
 - b. Assigning pictures to events/Allowing upload

2. Frontend
 - a. Viewing all data in a single page
 - b. Sign Up Page
 - c. Organizer Monitoring for volunteers
 - d. Creation page for events
 - e. User profile page

Known Issues

BACK END

1. Items are created without checking for token validation
2. Items are deleted without checking for token validation
3. Testing is has not been done on all functions
4. Searching descriptions/titles is broken for Slots, Events, and organizations

FRONT END

1. No front-end display for all features

Functionality

BACKEND

1. Slots
 - a. Read - /ProjectFolder/api/objects/slots/read/
 - i. By default will show you all slots in database
 - ii. Add '?id = #' to the url to get specific Slot with that ID
 - iii. Add '?page=#' to the url for 10(default) slots on the # page
 - iv. Add '?page=#&count=% ' to the url for % slots on the# page
 - b. Create - /ProjectFolder/api/objects/slots/create/
 - i. Post to page with variables (event_id, title, description, date, starttime, endtime, min, max)
 - c. Update - /ProjectFolder/api/objects/slots/update/
 - i. Post to page with variables (id, title, description, date, starttime, endtime, min, max)
 - d. Delete - /ProjectFolder/api/objects/slots/delete/
 - i. Post to page with 'id'
2. Events
 - a. Read - /ProjectFolder/api/objects/events/read/
 - i. By default will show you all events in database
 - ii. Add '?id = #' to the url to get specific Slot with that ID
 - b. Create - /ProjectFolder/api/objects/events/create/
 - i. Post to page with variables (title, description, email, phone, public)
 - c. Update - /ProjectFolder/api/objects/events/update/
 - i. Post to page with variables (id, title, description, email, phone, public)
 - d. Delete - /ProjectFolder/api/objects/events/delete/
 - i. Post to page with 'id'
3. Organizations

- a. Read - /ProjectFolder/api/objects/events/read/
 - i. By default will show you all slots in database
 - ii. Add '?id = #' to the url to get specific Slot with that ID
- b. Create - /ProjectFolder/api/objects/events/create/
 - i. Post to page with variables (title, description, email, phone, public)
- c. Update - /ProjectFolder/api/objects/events/update/
 - i. Post to page with variables (id, title, description, email, phone, public)
- d. Delete - /ProjectFolder/api/objects/events/delete/
 - i. Post to page with 'id'

FRONTEND

Viewing Events

Read Events

Search events...

Q

+ Create Event

id	title	description	email	phone	org_name	Action
4	CREATE Fall 2020	Let us show you what we worked on this Fall. Food is free too!	CREATE@ggc.edu	000-000-0001	Georgia Gwinnett College	<div><div> Read</div><div> Edit Event</div><div> Edit Slot</div><div> Delete</div></div>
3	Grad. Party	Party all day!!!!	ggcEvents@ggc.edu	888-888-8888	Georgia Gwinnett College	<div><div> Read</div><div> Edit Event</div><div> Edit Slot</div><div> Delete</div></div>
2	Car Show at Mall of Georgia	Bring your car and show it off to strangers you dont know. Why not its fun...	carShow@ggc.edu	123-456-7890	Test Organization	<div><div> Read</div><div> Edit Event</div><div> Edit Slot</div><div> Delete</div></div>

1

Read One Event

<input type="button" value="Read events"/>	
Event Id	5
Ororganization_ID	2
Event Title	Demo Create Event
Description	I have created this event to show that create events is working.
Email	CREATE@ggc.edu
Phone	890-345-6789
Job Title	null
Job Description	null
Job Start Time	null
Job End Time	null
Min Position	null
Max Position	null

Viewing Events (Code)

```
event_index.html JS read-events.js x
1 $(document).ready(function(){
2
3     // show list of product on first load
4     showEventsFirstPage();
5
6
7     // when a 'read Events' button was clicked
8     $(document).on('click', '.read-events-button', function(){
9         showEventsFirstPage();
10    });
11
12    // when a 'page' button was clicked
13    $(document).on('click', '.pagination li', function(){
14        // get json url any
15        var json_url=$(this).find('a').attr('data-page');
16
17        // show list of Events
18        showEvents(json_url);
19    });
20
21
22 });
23
24 function showEventsFirstPage(){
25     var json_url="http://localhost/Yietapi/event/read_paging.php";
26     showEvents(json_url);
27 }
28
29 // function to show list of Events
30 function showEvents(json_url){
31
32     // get list of Events from the API
33     $.getJSON(json_url, function(data){
34
35         // html for listing Events
36         readEventsTemplate(data, "");
37
38         // chage page title
39         changePageTitle("Read Events");
40
41     });
42 }
```

Create Events

Create Event

Read events

Organizer ID	Georgia Gwinnett College
Event Title	
Description	
Email	
Phone	
<div>+ Create event</div>	

Create Events (Code)

```
83
84 });
85     });
86
87     // will run if create event form was submitted
88     $(document).on('submit', '#create-event-form', function(){
89         // get form data
90         var form_data=JSON.stringify($(this).serializeObject());
91         // submit form data to api
92         $.ajax({
93             url: "http://localhost/Yietapi/event/create.php",
94             type : "POST",
95             contentType : 'application/json',
96             data : form_data,
97             success : function(result) {
98                 // event was created, go back to events list
99                 showEventsFirstPage();
100             },
101             error: function(xhr, resp, text) {
102                 // show error to console
103                 console.log(xhr, resp, text);
104             }
105         });
106
107         return false;
108     });
109 });
110 });
```


Update Events

Update Event

[Read events](#)

Organizer Name	Georgia Gwinnett College
Event Title	Demo Edit Event
Description	I have created this event to show that create events is working.
Email	CREATE@ggc.edu
Phone	890-345-6789
Update event	

Update Events (Code)

```
114
115     });
116
117     // will run if 'create event' form was submitted
118     $(document).on('submit', '#update-event-form', function(){
119
120         // get form data
121         var form_data=JSON.stringify($(this).serializeObject());
122
123         // submit form data to api
124         $.ajax({
125             url: "http://localhost/Yietapi/event/update.php",
126             type : "POST",
127             contentType : 'application/json',
128             data : form_data,
129             success : function(result) {
130                 // event was created, go back to events list
131                 showEventsFirstPage();
132             },
133             error: function(xhr, resp, text) {
134                 // show error to console
135                 console.log(xhr, resp, text);
136             }
137         });
138
139         return false;
140     });
141 });
```

Delete Events

Read Events

Search events...

Are you sure you want to delete this event?

No Yes

+ Create Event

id	title	description	email	phone	org_name	Action
5	Demo Edit Event	I have created this event to show that create events is working.	CREATE@ggc.edu	890-345-6789	Georgia Gwinnett College	Read Edit Event Edit Slot Delete
4	CREATE Fall 2020	Let us show you what we worked on this Fall. Food is free too!	CREATE@ggc.edu	000-000-0001	Georgia Gwinnett College	Read Edit Event Edit Slot Delete
3	Grad. Party	Party all day!!!!	ggcEvents@ggc.edu	888-888-8888	Georgia Gwinnett College	Read Edit Event Edit Slot Delete
2	Car Show at Mall of Georgia	Bring your car and show it off to strangers you dont know. Why not its fun...	carShow@ggc.edu	123-456-7890	Test Organization	Read Edit Event Edit Slot Delete

1

Delete Events (Code)

```
21     }
22 },
23 callback: function (result) {
24     if(result==true){
25
26         // send delete request to api / remote server
27         $.ajax({
28             url: "http://localhost/Yietapi/event/delete.php",
29             type : "POST",
30             dataType : 'json',
31             data : JSON.stringify({ id: event_id }),
32             success : function(result) {
33
34                 // re-load list of events
35                 showEventsFirstPage();
36             },
37             error: function(xhr, resp, text) {
38                 console.log(xhr, resp, text);
39             }
40         });
41     }
42 }
43
44
45
46 }
47 });
48
```

Testing Coverage

PROGRAM

Codeception was used for testing on the web application. Codeception is a multi-featured testing framework for PHP. It can handle unit, functional, and acceptance testing of web applications and it's powered by the already very popular PHPUnit testing framework.

Acceptance testing allows us to test our applications using the normal website viewing process of visit a webpage, fill in a form, and submit the form to see the desired result. The difference is with Codeception, we don't have to waste time going to the browser each time we want to test a new feature out, instead we can just run our acceptance tests to see if they pass or not.

<https://code.tutsplus.com/tutorials/acceptance-testing-with-codeception--net-36337>

The link above provides a thorough tutorial for installing and using Codeception.



Automated Testing Using **Codeception**

Read Test

```
apiEventReadCest.php x
1  <?php
2  use Codeception\Util\JsonArray;
3
4
5  class apiCest
6  {
7      #===== READ TEST =====||
8
9      public function doesID4Exist(ApiTester $I)
10     {
11         $I->sendGet('/event/read.php');
12         $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
13         $I->seeResponseIsJson();
14         $I->seeResponseContainsJson(['id' => '2']);
15     }
16
17     public function doesGGCOrgnameExist(ApiTester $I)
18     {
19         $I->sendGet('/event/read.php');
20         $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
21         $I->seeResponseIsJson();
22         $I->seeResponseContainsJson(['org_name' => 'Georgia Gwinnett College']);
23     }
24
25     public function isPhoneNumber911Used(ApiTester $I)
26     {
27         $I->sendGet('/event/read.php');
28         $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
29         $I->seeResponseIsJson();
30         $I->seeResponseContainsJson(['phone' => '911']);
31     }
32
33     public function checkForEventByEmail(ApiTester $I){
34         $I->sendGet('/event/read.php');
35         $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
36         $I->seeResponseIsJson();
37         $I->seeResponseContainsJson(['email' => 'SoccerTeam@ggc.edu']);
38     }
39
40 }
```

```
api.suite.yml x
1  class_name: ApiTester
2  modules:
3      enabled:
4          - \Helper\Api
5          - Asserts
6          # Add the lines below
7          - REST:
8              # Add your base API URL
9              url: http://localhost/Yietapi/
10             # Can also be a framework module name
11             depends: PhpBrowser
12             # Limits PhpBrowser to JSON or XML
13             part: Jsonl
```

```

public function doesID4Exist(ApiTester $I)
{
    $I->sendGet('/event/read.php');
    $I->canSeeResponseCodeIs(\Codeception\Util\StatusCode::OK);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(['id' => '2']);
}

public function doesGGCOrnameExist(ApiTester $I)
{
    $I->sendGet('/event/read.php');
    $I->canSeeResponseCodeIs(\Codeception\Util\StatusCode::OK);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(['org_name' => 'Georgia Gwinnett College']);
}

```

```

public function isPhoneNumber911Used(ApiTester $I)
{
    $I->sendGet('/event/read.php');
    $I->canSeeResponseCodeIs(\Codeception\Util\StatusCode::OK);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(['phone' => '911']);
}

public function checkForEventByEmail(ApiTester $I){
    $I->sendGet('/event/read.php');
    $I->canSeeResponseCodeIs(\Codeception\Util\StatusCode::OK);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(['email' => 'SoccerTeam@ggc.edu']);
}

```

```

Nataanaelz-MBP:TestingFolder nataanaelvnistor$ t run api apiEventReadCest.php:checkForEventByEmail
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✓ apiCest: Check for event by email (0.02s)

Time: 81 ms, Memory: 10.00MB

OK (1 test, 4 assertions)
Nataanaelz-MBP:TestingFolder nataanaelvnistor$

```

```

Nataanaelz-MBP:TestingFolder nataanaelvnistor$ t run api apiEventReadCest.php:doesID4Exist
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✓ apiCest: Does id4 exist (0.04s)

Time: 214 ms, Memory: 10.00MB

OK (1 test, 4 assertions)
Nataanaelz-MBP:TestingFolder nataanaelvnistor$

```

```

Nataanaelz-MBP:TestingFolder nataanaelvnistor$ t run api apiEventReadCest.php:doesGGCOrnameExist
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✓ apiCest: Does ggc orname exist (0.02s)

Time: 81 ms, Memory: 10.00MB

OK (1 test, 4 assertions)
Nataanaelz-MBP:TestingFolder nataanaelvnistor$

```

```

Nataanaelz-MBP:TestingFolder nataanaelvnistor$ t run api apiEventReadCest.php:isPhoneNumber911Used
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✗ apiCest: Is phone number911 used (0.02s)

Time: 83 ms, Memory: 10.00MB

There was 1 failure:

1) apiCest: Is phone number911 used
Test tests/api/apiEventReadCest.php:isPhoneNumber911Used
Step See response contains json ("phone": "911")
Fail Response JSON does not contain the provided JSON

- Expected | + Actual
ee ee
Array (
- 'phone' => '911'
+ 'records' => Array (...)
)

Scenario Steps:

4. $I->seeResponseContainsJson(["phone": "911"]) at tests/api/apiEventReadCest.php:30
3. $I->seeResponseIsJson() at tests/api/apiEventReadCest.php:29
2. $I->canSeeResponseCodeIs(200) at tests/api/apiEventReadCest.php:28
1. $I->sendGET("/event/read.php") at tests/api/apiEventReadCest.php:27

FAILURES!
Tests: 1, Assertions: 4, Failures: 1.

```

Create/Delete Test

```
apiCreate_DeleteCest.php x
1  <?php
2  use Codeception\Util\JsonArray;
3
4  class apiCreate_DeleteCest
5  {
6      #===== CREATE TEST =====||
7
8      public function createFakeEvent(ApiTester $I){
9          $I->haveHttpHeader('Content-Type', 'application/json');
10         $I->sendPOST('/event/create.php',
11             [
12                 'id' => '000',
13                 'organization_id' => '1',
14                 'org_name' => 'Te Loading... zation',
15                 'title' => 'Automated Test CREATE',
16                 'description' => 'Added by the automated test Codeception',
17                 'email' => 'ggcFakeEventTest@ggc.edu',
18                 'phone' => '098-765-4321',
19                 'job_description' => null,
20                 'Job Start Time' => null,
21                 'Job End Time' => null,
22                 'max_position' => null,
23                 'min_position' => null
24             ]);
25         $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::CREATED);
26         $I->seeResponseIsJson();
27         $I->seeResponseContainsJson(['message' => 'event was created.']);
28     }
29
30     #===== READ CREATED TEST =====||
31     public function checkToSeeCreatedExists(ApiTester $I){
32         $I->sendGet('/event/read.php');
33         $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
34         $I->seeResponseIsJson();
35         $I->seeResponseContainsJson(
36             [
37                 'organization_id' => '1',
38                 'org_name' => 'Test Organization',
39                 'title' => 'Automated Test CREATE',
40                 'description' => 'Added by the automated test Codeception',
41                 'email' => 'ggcFakeEventTest@ggc.edu',
42                 'phone' => '098-765-4321',
43                 'job_description' => null,
44                 'Job Start Time' => null,
45                 'Job End Time' => null,
46                 'max_position' => null,
47                 'min_position' => null
48             ]);
49
50         $id = $I->grabDataFromResponseByJsonPath('$.records[0].id[]');
51         $deleteID = $id[0];
52         echo "===== ".$deleteID;
53     }
54 }
55
56 #===== DELETE CREATED TEST =====||
57 public function deleteCreatedEntry(ApiTester $I){
58     $I->sendGet('/event/read.php');
59     $id = $I->grabDataFromResponseByJsonPath('$.records[0].id[]');
60     $deleteID = $id[0];
61     echo "===== ".$deleteID;
62     $I->haveHttpHeader('Content-Type', 'application/json');
63     $I->sendPOST('/event/delete.php',
64         [
65             'id' => $deleteID
66         ]);
67     $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
68     $I->seeResponseIsJson();
69     $I->seeResponseContainsJson(['message' => 'Event was deleted.']);
70 }
71
```



```

public function createFakeEvent(ApiTester $I){
    $I->haveHttpHeader('Content-Type', 'application/json');
    $I->sendPOST('/event/create.php',
        [
            'id' => '000',
            'organization_id' => '1',
            'org_name' => 'Test Organization',
            'title' => 'Automated Test CREATE',
            'description' => 'Added by the automated test Codeception',
            'email' => 'ggcFakeEventTest@ggc.edu',
            'phone' => '098-765-4321',
            'job_description' => null,
            'Job Start Time' => null,
            'Job End Time' => null,
            'max_position' => null,
            'min_position' => null
        ]
    );
    $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::CREATED);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(['message' => 'event was created.']);
}

```



```

TestingFolder — -bash — 152x48
Natanaelz-MBP:TestingFolder natanaelnistor$ t run api apiCreate_DeleteCest:createFakeEvent
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✓ apiCreate_DeleteCest: Create fake event (0.02s)
-----

Time: 84 ms, Memory: 10.00MB

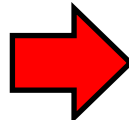
OK (1 test, 4 assertions)
Natanaelz-MBP:TestingFolder natanaelnistor$

```

```

public function deleteCreatedEntry(ApiTester $I){
    $I->sendGet('/event/read.php');
    $id = $I->grabDataFromResponseByJsonPath('$.records[0].id[]');
    $deleteID = $id[0];
    echo "===== ".$deleteID;
    $I->haveHttpHeader('Content-Type', 'application/json');
    $I->sendPOST('/event/delete.php',
        [
            'id' => $deleteID
        ]
    );
    $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(['message' => 'Event was deleted.']);
}

```



```

TestingFolder — -bash — 152x48
Natanaelz-MBP:TestingFolder natanaelnistor$ t run api apiCreate_DeleteCest:deleteCreatedEntry
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✓ apiCreate_DeleteCest: Delete created entry (0.02s)=====24
-----

Time: 86 ms, Memory: 10.00MB

OK (1 test, 4 assertions)
Natanaelz-MBP:TestingFolder natanaelnistor$

```

```

public function checkToSeeCreatedExists(ApiTester $I){
    $I->sendGet('/event/read.php');
    $I->canSeeResponseCodeIs(\Codeception\Util\HttpCode::OK);
    $I->seeResponseIsJson();
    $I->seeResponseContainsJson(
        [
            'organization_id' => '1',
            'org_name' => 'Test Organization',
            'title' => 'Automated Test CREATE',
            'description' => 'Added by the automated test Codeception',
            'email' => 'ggcFakeEventTest@ggc.edu',
            'phone' => '098-765-4321',
            'job_description' => null,
            'Job Start Time' => null,
            'Job End Time' => null,
            'max_position' => null,
            'min_position' => null
        ]
    );

    $id = $I->grabDataFromResponseByJsonPath('$.records[0].id[]');
    $deleteID = $id[0];
    echo "===== ".$deleteID;
}

```



```

TestingFolder — -bash — 152x48
Natanaelz-MBP:TestingFolder natanaelnistor$ t run api apiCreate_DeleteCest:checkToSeeCreatedExists
Codeception PHP Testing Framework v2.5.1
Powered by PHPUnit 7.3.5 by Sebastian Bergmann and contributors.
Running with seed:

Api Tests (1) -----
✓ apiCreate_DeleteCest: Check to see created exists (0.02s)=====24
-----

Time: 84 ms, Memory: 10.00MB

OK (1 test, 4 assertions)
Natanaelz-MBP:TestingFolder natanaelnistor$

```

Team Members

PROJECT INTRODUCTION

Student Michael Holtmann (mikedh87@gmail.com, (404) 431-8272) wants a website for volunteer and event tracking. User stories: I would like a web based software that will help with the origination of volunteers across many events. I need the ability to place, on a calendar, an event and break it into time slots and number of volunteers needed. I my volunteers to have the ability to register for a specific time slot for a specific day and specific event. I need the system to track who signed up to help for a specific time slot for a specific day and specific event.

Rules:

1. Teams must be at least 3 and up to 4 people. Exception; to be a 5-person team, you really need to justify why you need so many people. That doesn't mean "oh this project is so big!", you have to clearly separate the work into different members.
2. You're free to come up with ideas, but you need to make sure your client accepts your plan. You can always bring more external clients to the class.
3. Each person must have one or multiple of the roles below.

Team Roles

1. Code architecture/lead programmer: framework, components, coding style. Doesn't mean the only programmer! Everyone must program, lead programmer decides on the overall design and coding standards.
2. UI/UX design: User Interface and User experience. Make mock UI designs and user-friendly navigation.
3. Data modeler: Database, data files, server-client structure, data structures.
4. Team manager: Sprint planning, and time accounting for the team.
5. Client liaison: Communications, demos, and negotiations with client, what was done and planned to handle client's requirements (role can be combined with manager)
6. Documentation lead: Setup documentation conventions for code and project, then delegate work onto teammates. Manage presentations, reports, and licensing documentation. Everyone must add documentation!
7. Testing lead: Setup testing framework and conventions. Use continuous integration to test at each commit/push. Everyone must do their own testing!



Ammar Huseinspahic

Code Architecture / Lead Programmer & Testing Lead

Concentration: System Security

Year: Graduating Senior

Languages: PHP, CSS, JavaScript, C#, Java

Programs: Unity, PhotoShop, WAMP

Past Software Projects:

- Software Dev 1: 2d web game featuring dynamic game world, sounds, and multiplayer
- Info Project: Study Room reservation system with online tracking of room availability
- Website for local business made with BootStrap
- Lots of personal side projects



Yiet Mai
Data Modeler & UI/UX Design

Year: May, 2019

Skills (computer languages, platforms, etc)

- Java
- Javascript
- PHP
- HTML & CSS
- SQL & mySQL

Interests (keep it relevant)

- Designing and building web
- Software Development
- Game development

Expertise in software development tools (IDEs, version control, build tools, etc)

- IDE's: Eclipse
- Sublime
- Netbean
- Atom

Past software projects (briefly explain the scope and your role)

- Software Development I Text Base Adventure Game.
- Projects from Web Dev.
- Projects from Digital Media



Natanael V. Nistor

Team Manager, Client Liaison, & Documentation Lead

Concentration: Software Dev

Year: Senior

Languages:

- Java
- JavaScript
- jQuery
- SQL
- Html/CSS
- React
- PHP

IDE / Editors: Eclipse, Atom.io, VS Code, IntelliJ, Note Pad ++, Android Studio, Unity

Interests:

- Coding ...
- AI
- Web Dev
- Games Dev
- Drones

Projects:

- Text Based Game (Java)
- Small Web Technology websites projects
- Other Class Homework and Assignments

Survey

GGC Fall 2018 - ITEC3870

Event Tracking System



Web based event tracking built
for organizers and volunteers.

Ammar Huseinspahic || Yiet Mai || Natanael Nistor

Your honest feedback is important to us!

Start

press ENTER

<https://evtsys2018.typeform.com/to/XtUneI>

EVENT TRACKER
INTELLECTUAL PROPERTY AND LICENSING AGREEMENT
[Creative Commons Attribution Non-Commercial (CC-BY-NC)]

This Intellectual Property Contribution and Assignment Agreement (the "Agreement") is made as of 22 November 2018 by and between the members of Event Tracker (the "Company"), Michael Holtmann (the "Client"), and Dr. Cengiz Gunay (the "Sponsor").

1. Intellectual Property Assignment

BOTH PARTIES HAVE AGREED THAT THE MEMBERS OF COMPANY WILL EACH RECEIVE TEN PERCENT OF INTELLECTUAL CLAIM, THE CLIENT WOULD BE ASSIGNED SIXTY-FIVE PERCENT OF INTELLECTUAL CLAIM, AND THE SPONSOR WOULD BE ASSIGNED FIVE PERCENT OF INTELLECTUAL CLAIM. INTELLECTUAL CLAIM INCLUDES ENTIRE RIGHTS, TITLE AND INTEREST IN AND TO ANY AND ALL OF THE PROPERTIES THAT EXIST AS OF THE DATE HEREOF.

2. Licensing Agreement

THE WORK (EVENT TRACKER) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

3. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

4. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

5. Termination

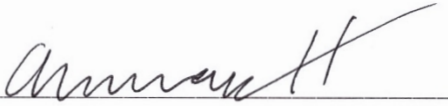
THIS LICENSE AND THE RIGHTS GRANTED HEREUNDER WILL TERMINATE AUTOMATICALLY UPON ANY BREACH BY YOU OF THE TERMS OF THIS LICENSE. INDIVIDUALS OR ENTITIES WHO HAVE RECEIVED ADAPTATIONS OR COLLECTIONS FROM YOU UNDER THIS LICENSE, HOWEVER, WILL NOT HAVE THEIR LICENSES TERMINATED PROVIDED SUCH INDIVIDUALS OR ENTITIES REMAIN IN FULL COMPLIANCE WITH THOSE LICENSES.

IN WITNESS, WHEREOF, THE PARTIES HERE TO THE DRAWN AGREEMENT AND WILL BE EXECUTED THE DAY AND YEAR FIRST WRITTEN ABOVE.

PLEASE SIGN BELOW:

12/4/2018

Date



Ammar Huseinspahic - Code Architecture/Lead Programmer & Testing

12/4/2018

Date



Natanael Nistor - Team manager & Documentation Lead

12/4/18

Date



Yiet Mai - Data modeler & UI/UX Design

12/4/18

Date



Michael Holtmann - Client

Date

Cengiz Gunay - Sponsor