Heart Screening Management System – Milestone 1

Team Members
Carlo Campanini
Chris Newberry
Drew Dunkelberger
John Dewey
Noah Wilson

Client
Evan Ernst, CEO
Klynton Holmes, Tech Advisor

Faculty Sponsor
Dr. Eraldo Ribeiro

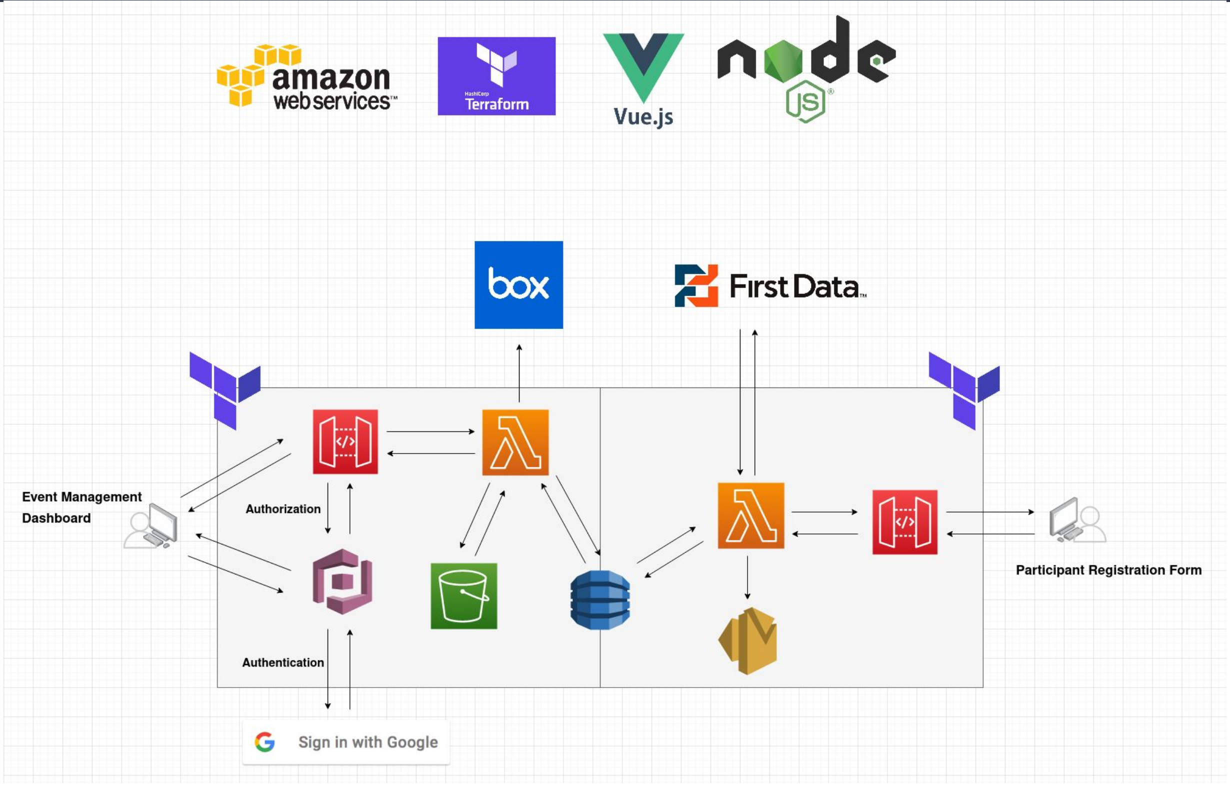
Milestone 1 Overview

- Provide a demo using the selected AWS tools
- Provide demo for test tools, Mocha and Chai
- Resolved technical challenges
- Selected collaboration tools
  - Gitlab, Github, Slack, Discord, Google Drive
- Created a requirements, design, and test document
- Began implementing/testing features:
  - Email reminder notification day before event
  - Close event registration day before event
  - Vary verbiage depending on participant’s age
  - Auto display available times slots for private events
  - Provide link for private events

Functional Requirements

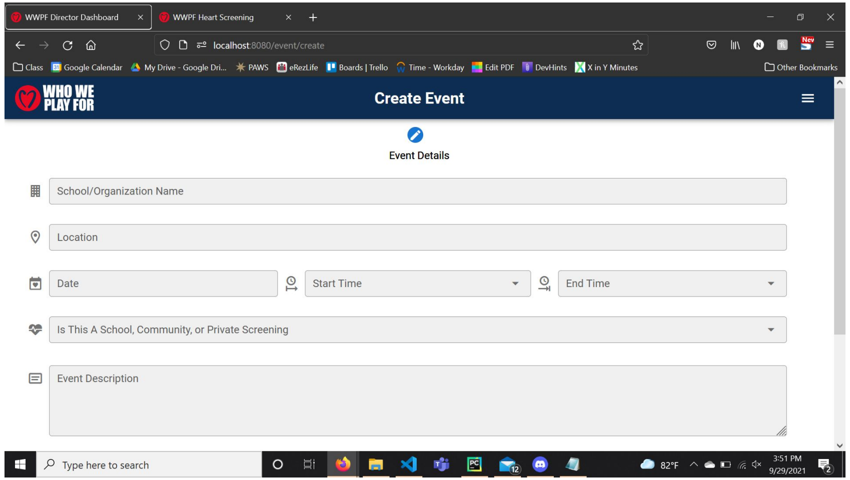
- Private/restricted event functionality with shareable link
- Auto display event time slots after private event creation
- Email reminders and finalize registration for events the day before
- Varying verbiage depending on participant’s age
- Accountant ability to search/filter payments and export to spreadsheet
- Participant ability to cancel previous registration
- Director ability to request deletion of a previously created event
- Generation of QR codes for on-site event registration

System Architecture



User Interface – Director

- Create and publish events:



WWPF Director Dashboard | WWPF Heart Screening

localhost:8080/event/create

Class | Google Calendar | My Drive - Google Dri... | PAWS | eRezLife | Boards \| Trello | Time - Workday | Edit PDF | DevHints | X in Y Minutes | Other Bookmarks

WHO WE PLAY FOR

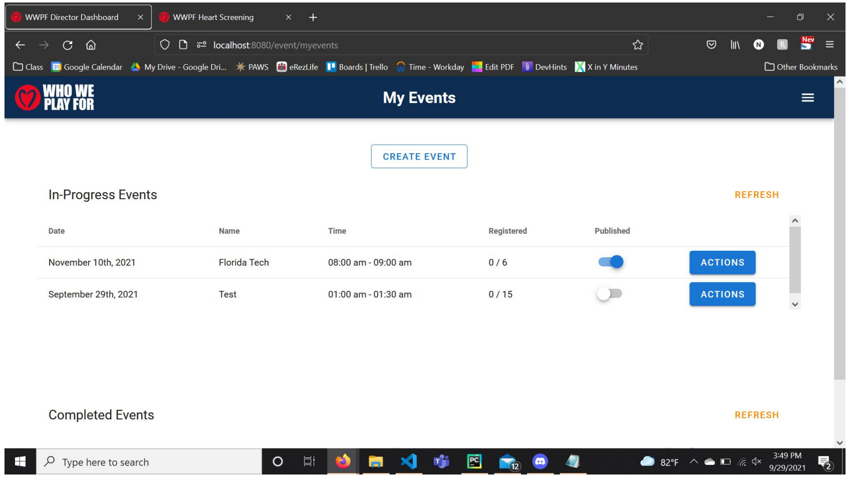
Create Event

Event Details

School/Organization Name
Location
Date / Start Time / End Time
Is This A School, Community, or Private Screening
Event Description

Type here to search | 82°F | 3:51 PM 9/29/2021

- View your in-progress events and completed events:



WWPF Director Dashboard | WWPF Heart Screening

localhost:8080/event/myevents

Class | Google Calendar | My Drive - Google Dri... | PAWS | eRezLife | Boards \| Trello | Time - Workday | Edit PDF | DevHints | X in Y Minutes | Other Bookmarks

WHO WE PLAY FOR

My Events

CREATE EVENT

In-Progress Events — REFRESH

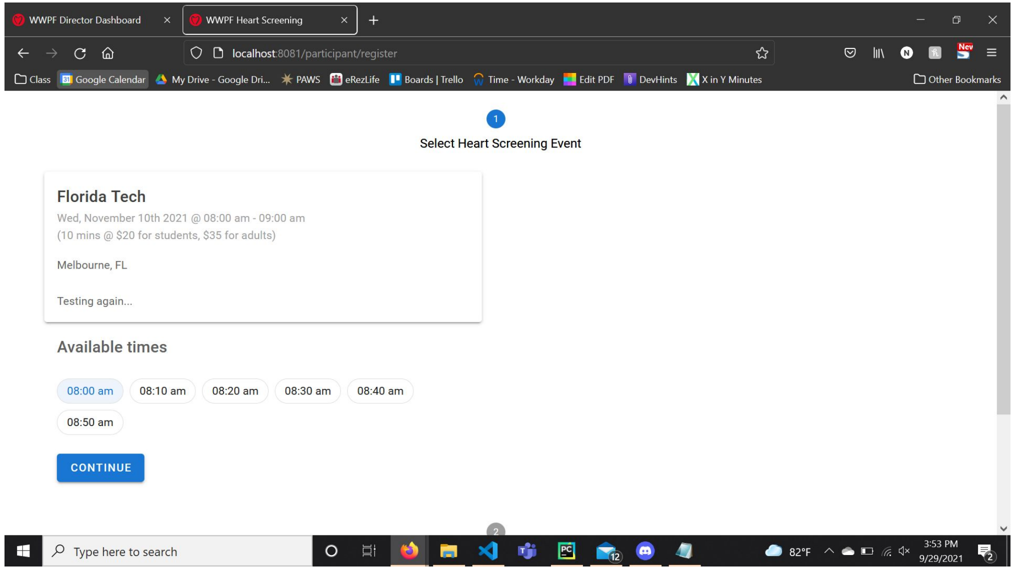
Date	Name	Time	Registered	Published	
November 10th, 2021	Florida Tech	08:00 am - 09:00 am	0 / 6	●	ACTIONS
September 29th, 2021	Test	01:00 am - 01:30 am	0 / 15	○	ACTIONS

Completed Events — REFRESH

Type here to search | 82°F | 3:49 PM 9/29/2021

User Interface – Participant

- View published events and register for them:



1

Select Heart Screening Event

Florida Tech

Wed, November 10th 2021 @ 08:00 am - 09:00 am

(10 mins @ $20 for students, $35 for adults)

Melbourne, FL

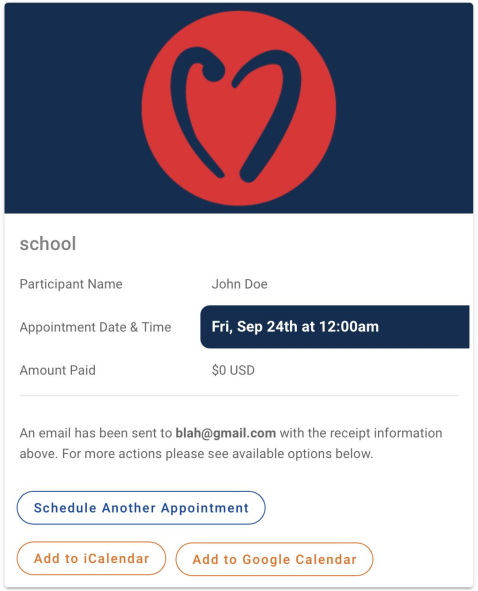
Testing again...

Available times

08:00 am | 08:10 am | 08:20 am | 08:30 am | 08:40 am | 08:50 am

CONTINUE

- Confirmation of registration and calendar integration:

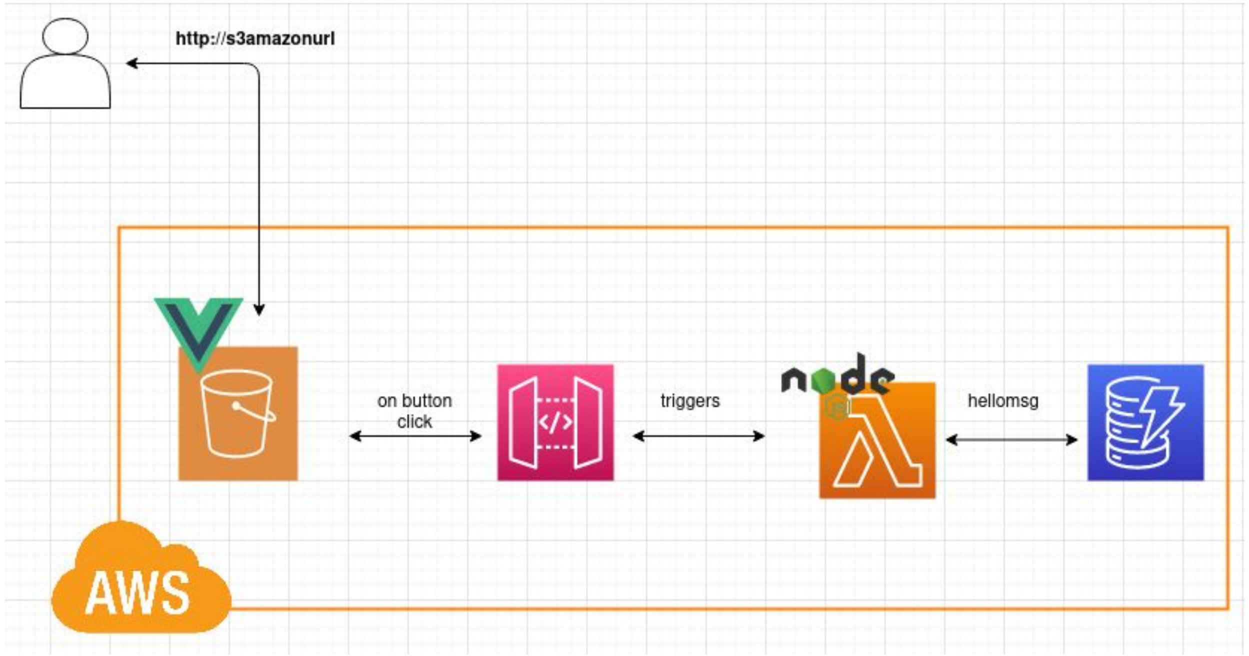


school

Participant Name	John Doe
Appointment Date & Time	Fri, Sep 24th at 12:00am
Amount Paid	$0 USD

An email has been sent to **blah@gmail.com** with the receipt information above. For more actions please see available options below.

Hello World Web Application Demo



Hello world app link:
http://hellobucketwwpf.s3-website.us-east-2.amazonaws.com

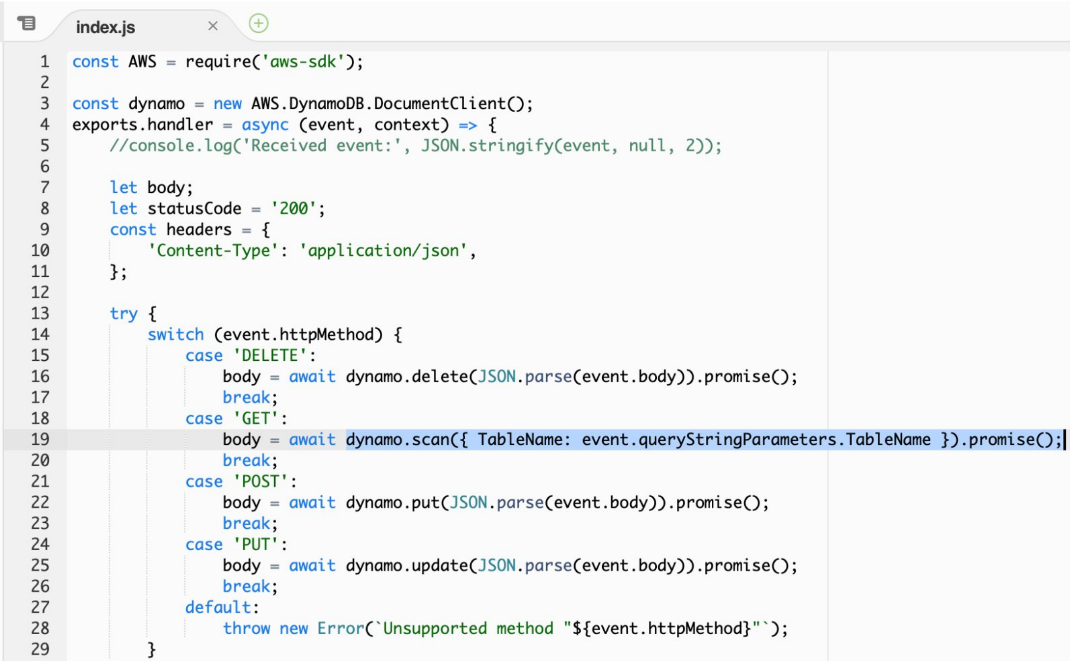
Hello World Web Application Demo

Lambda Function:

- API Gateway triggers on button click (GET request)

DynamoDB Table containing message:

- Lambda function uses a scan() to get table contents

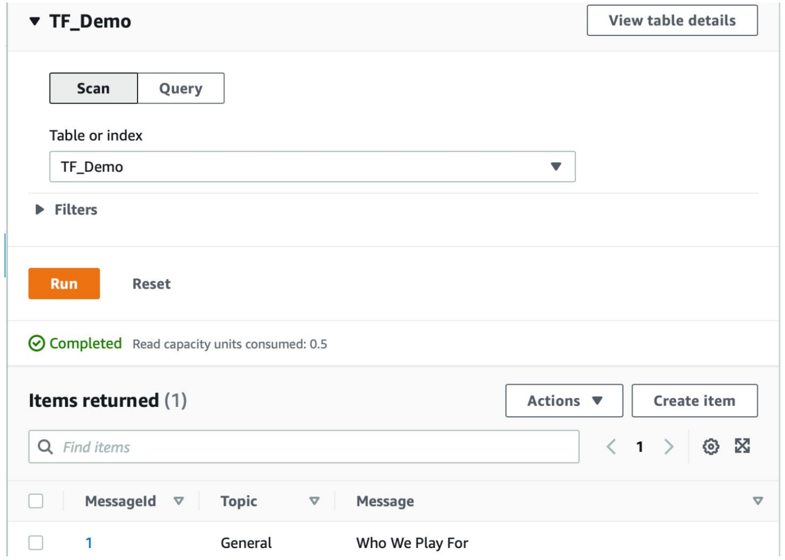


```javascript
const AWS = require('aws-sdk');

const dynamo = new AWS.DynamoDB.DocumentClient();
exports.handler = async (event, context) => {
    //console.log('Received event:', JSON.stringify(event, null, 2));

    let body;
    let statusCode = '200';
    const headers = {
        'Content-Type': 'application/json',
    };

    try {
        switch (event.httpMethod) {
            case 'DELETE':
                body = await dynamo.delete(JSON.parse(event.body)).promise();
                break;
            case 'GET':
                body = await dynamo.scan({ TableName: event.queryStringParameters.TableName }).promise();
                break;
            case 'POST':
                body = await dynamo.put(JSON.parse(event.body)).promise();
                break;
            case 'PUT':
                body = await dynamo.update(JSON.parse(event.body)).promise();
                break;
            default:
                throw new Error(`Unsupported method "${event.httpMethod}"`);
        }
```



Completed Read capacity units consumed: 0.5

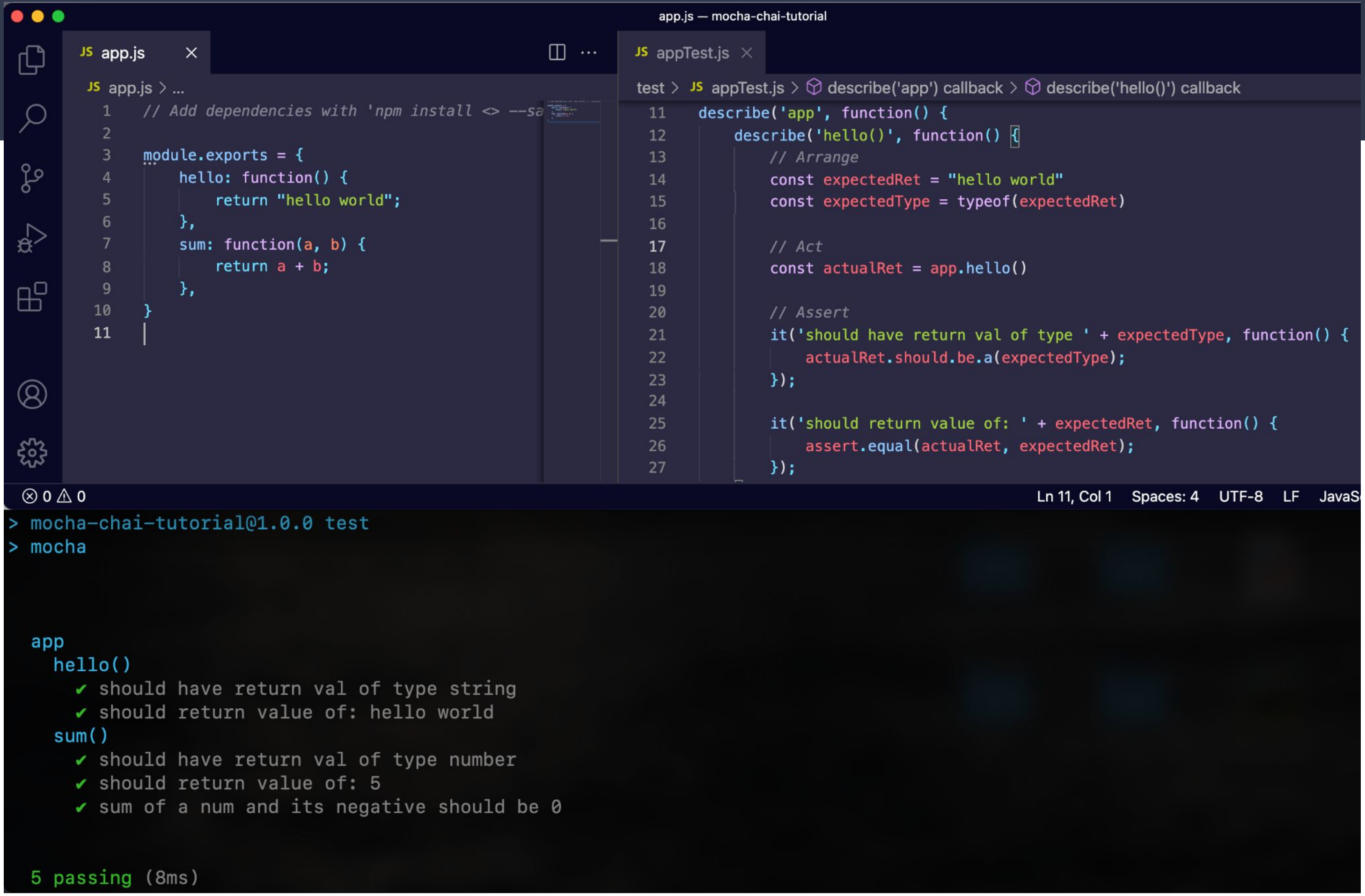
Items returned (1)

MessageId	Topic	Message
1	General	Who We Play For

Testing

- Team members will be following the test driven development (TDD) methodology
  - Write tests before any code
  - AAA - Arrange, Act, Assert
- Each feature will have several test cases, which consider common situations and edge cases

Mocha and Chai Test Demo



app.js — mocha-chai-tutorial

app.js

```
// Add dependencies with 'npm install <> --sa

module.exports = {
    hello: function() {
        return "hello world";
    },
    sum: function(a, b) {
        return a + b;
    },
}

```

appTest.js

test > appTest.js > describe('app') callback > describe('hello()') callback

```
describe('app', function() {
    describe('hello()', function() {
        // Arrange
        const expectedRet = "hello world"
        const expectedType = typeof(expectedRet)

        // Act
        const actualRet = app.hello()

        // Assert
        it('should have return val of type ' + expectedType, function() {
            actualRet.should.be.a(expectedType);
        });

        it('should return value of: ' + expectedRet, function() {
            assert.equal(actualRet, expectedRet);
        });
```

Ln 11, Col 1   Spaces: 4   UTF-8   LF   JavaS

```
> mocha-chai-tutorial@1.0.0 test
> mocha



  app
    hello()
      ✔ should have return val of type string
      ✔ should return value of: hello world
    sum()
      ✔ should have return val of type number
      ✔ should return value of: 5
      ✔ sum of a num and its negative should be 0


  5 passing (8ms)
```

Milestone 2 Tasks
(in order of priority)

Implement, test, and demo:

1. In-progress tasks from milestone 1
2. Onsite screening results management with Cardea software
3. Searching payment by several filters
4. Exporting payments to spreadsheet
5. Participant cancel registration
6. Confirming/requesting deletion of event

Note: Tasks 2-6 will be distributed on a first-come-first-served basis to team members following the completion of task 1

Questions?