FADDY MICHEL

SENIOR WEB DEVELOPER
faddy.michel@gmail.com / +1-347-610-0502 / https://github.com/teatro13

As a Software Engineer with strong Computer Science skills, I have the ability to play the role of a Joker in a team; the person who can perform efficiently in different positions. This is achieved by understanding the fundamentals of the various paradigms of programming languages—Imperative (e.g., C), Object-Oriented (e.g., Java and JavaScript), Functional (e.g., Haskell and JavaScript), Logical (e.g., Prolog), Data-Driven (e.g., awk), etc. This in-depth understanding of the power of languages to manipulate the machine lead to the ability to work in any kind of project; leveraging any stack of technologies or architectures.

Facts

ak standks for Advanced knowledge, ik for Intermediate Knowledge and bk for Basic Knowledge.

Programming: bash (ak), GNU coreutils (ak), C (ik), Java (ak), JavaScript (ak), Python (ik), awk (bk), Perl (bk), SQL (ik), Csound (ik), lilypond (ak), HTML (ak), CSS (ak), LaTeX (ak), M4 (bk), sed (bk), prolog (ik)

Technologies: Git (ak), Unix/Linux (ak), Apache Maven (ik), Gradle (ik), NodeJS (ak), Google Cloud platform (ik), AWS (ik), Adobe CQ/AEM (ik), KDemandware (ik), NetSuite (ik), MySQL (ik), Apache Cassandra (ik), MongoDB (ik), Spring (bk), Grunt (bk), Gulp (bk), Chrome Extensions (ik)

Operating Systems: GNU Linux/Debian (ak), MacOS (ak), Ubuntu (ak), Windows (bk), Android (ik), iOS (bk), JoliOS (bk), Chrome OS (ak), Firefox OS (bk)

Employment History

OCT 2018--PRESENT**Founder / Leader / Software Engineer at Teatro13**

Founded Workshop for making not only ACCESSIBLE but USABLE Technologies.

Teatro: (https://github.com/teatro13/teatro) It all starts from the Playwriter; who must have programming skills along with creativity. The Playwriter writes different Scenarios of a Play to be hosted on Teatro so that each Scenario focuses on a specific type of Participants. A Scenario is a JavaScript function running on the NodeJS runtime with a Participant and a Ticket passed as parameters. After writing and hosting Scenarios, Tickets can be issued for a Scenario where Ticket is the Accessibility method for a Participant to participate in a Play. Participants (Clients which can be a Web Browser such as Chrome and Firefox, Mobile App running on iOS or Android, etc) connect to Teatro (Server) through the WebSocket protocol of communication (Read about WebSocket on [Wikipedia](https://en.wikipedia.org/wiki/WebSocket)). Since WebSocket opens a separate channel between the Server and each Client, making it possible for both sides to send/receive data to/from the other side at any time, a live experience can be built step-by-step between the Participant and the playing Scenario. This way, at each step, the Participant can translate the received data in a format (audible, graphical, etc.) that suits their physical capabilities and the situation they are facing. This provides a smoother User-Experience than the Apps of nowadays since they depend on loading a lot of User-Interface components all together at once, providing a crowded User-Experience, which makes it hard to translate all of these components into different formats in a smooth way (Apps made by Google, Microsoft, Apple, Facebook, Amazon, LinkedIn and all of their Blind followers are obvious examples of such a crowded and poor User-Experience). In the end, Teatro aims to move a step forward towards Equality in Usability that is believed to start from the backend before the frontend.

Scenarist: (https://github.com/teatro13/scenarist) A Scenario-based UI framework for web browsers. A Scenarist application consists of scenarios; where each scenario is built from scenes. A scene is an ES6 native module exporting setting object, establishment function and characters object. Setting objects exported by all scenes of a scenario are appended to each other forming a single setting object shared between all scenes of a scenario. Establishment function of each scene is run after loading scenes with thisArg (this keyword) set to the setting object of the scenario. Characters object owns two properties; events and action. Events is an array of scene names where the characters' action is executed whenever any of these events is triggered. The display property of the scenarist sets which scenario is currently in effect. Finally, an event is triggered by calling the play function of a scenarist passed with the scene name (event), character name and an optional parameter action representing the value which the character should be set to.

Oscilla: (https://github.com/teatro13/oscilla) A Sound Synthesizer for web browsers. It uses Scenarist to build the UI and Web Audio API for it's logic. In each scenario of Oscilla, ASCII codes are considered as the scenario characters. For instance, when on the Note scenario, the 's' ASCII code triggers the character responsible for playing the C pitch, and 'g' triggers the character responsible for decrementing the octave, and so forth.

`AUG 2015--MAR 2016`**Senior Software Engineer at Poppin**

At Poppin, there was three different technology-stacks. Here s the work done within each stack:

1. NetSuite, Business Management Software, led the development of:
   - Re-engineering and putting down the architecture of the code-base.
   - Developing a new model for selecting warehouse locations for each sales order.
   - Modifying and changing the data models for companies, contacts and customers, and the relationships between them, by writing a variety of scripts to migrate the current data.
   - Writing a service responsible for exporting records and sublists as CSV.
   - Bug fixing.
2. Demandware, E-commerce Platform, contributed to:
   - Re-designing Poppin's web application.
   - Modified the categorization model of products and making better use of the slicing attributes.
   - Bug fixing.
3. Scruffy, an In-house Java-based Feeds Management System, contributed to:
   - Refactoring customers feeds.
   - Bug fixing.

`MAR 2015--MAY 2015`**Senior Software Engineer at PM Digital**

Contributed to various projects for HP, ColorLok and KPMG. Focused on the front-end by helping adding and creating new UI modules and components. Technologies used include Adobe AEM/CQ, NodeJS, Grunt, AngularJS, Handlebars, SASS and LESS.

`JAN 2013--DEC 2014`**Software Engineer at The Daily Beast / Newsweek**

Developed a toolbar for old Newsweek Editions deployed to Adobe Content View on Kindles and Android devices: The toolbar gives the reader the ability to increase and decrease text size and save the last size using HTML5 storage, scroll to the top of the page and share on the top 3 social networks.

This project was challenging in the sense that it was developed to support old Android devices (2.3) which do not support various CSS properties, especially CSS3 properties and the fixed positioning do not work properly.

Contributed on the redesign of Newsweek, The Daily Beast and Women in the World (Desktop and mobile):

- Modifying and sometimes reimplementing the JSON content API which is used then by the Dust templating engine to render content.
- Created various UI components to be reused on different pages all over the sites. To achieve this, various front-end technologies and techniques were used. DustJS to create templates, Less for styling, Javascript and various in-house and third-party libraries to build dynamic features.
- Used Google Publisher Tags, Double-click for Publishers and Google AdSense to deliver Ads.
- Used Adobe SiteCatalyst, Dax and Google Analytics for tracking.

Developed a Java Service that compiles Less files on the server-side: This is done by checking for updated and newly created less files in JCR, using OSGI events, and then compile them using Mozilla Rhino. And finally save the CSS output in the same directory where each less file is placed. To optimize the performance of the service, used the multi-threading features provided by Java.

Developed a bundle that generates the printed version of Newsweek editions: The front-end templates where created using XSL-FO. A Java servlet with a xsl selector and PDF extension outputs the pdf version of the requested page. The PDF generation is done using a Java service that uses Apache FOP.

Migrated the legacy advertising library to use Google Publisher Tags: Created a JQuery plugin that is responsible for injecting the ad inside the selected elements. Also, created a Javascript library that uses Google Publisher Tags for ad calls.

`JAN 2011--NOV 2011`**Junior Software Engineer at TekTrust**

1. Took a training on Business Process Management using Appian BPM Suite to:
   - Design and model processes using Appian Process Modeler.
   - Design applications using Appian Applications designer.
   - Design forms using Appian forms designer.
   - Customize smart services, expression functions, custom forms and custom channels using Appian java APIs, JSP, HTML, Javascript, CSS and other web technologies.
2. As a member of three members team, built now kora social network connecting soccer players using Elgg open source social networking engine (MVC framework based on LAMP):
   - Designed and implemented an Elgg plugin for linking the users now kora account with his facebook account using the facebook graph API, it allows the user to login using facebook and updates his timeline with his now kora feeds.
   - Designed and implemented a plugin for organizing soccer matches between players in the network. The plugin allows a teams admin to create/edit/delete matches, invite team players to created matches and accept/reject non-team players join requests.

- Contributed in the Design and implementation of an Elgg web theme for all the pages the pages that appear when there is no user logged in using HTML4, CSS2 and jQuery UI.
- Designed and implemented two mobile Elgg views, one for touch devices using Sencha Touch 2 and the other for blackberry devices using JQuery Mobile. During this task, I had to customize a lot of javascript, jQuery, HTML5 and CSS3 code to build the UI as required by the client.

AUG 2008--NOV 2008**Intern Software Engineer at Atlantic Industries — The Cocacola Company**

Implemented web applications for the Business Unit Telecommunication cost management and reporting: The system was implemented using PHP and MSSQL.

Education

B.Sc., Computer Science and Engineering: 2005–2010 at German University in Cairo.