Dynamic Websites using Real Time Data

Introduction

Websites are used for many purposes including resumes, restaurants, visualizing data, and streaming videos. The methods and languages used for websites vary depending on their purpose. Modern technology allows data to be streamed continuously from a server to an endpoint, and the website updates as it receives the data. This paper reviews two types of websites and the technology used to gather and output data.

Websites using HTML, CSS, JavaScript, and Python

Static/ Traditional

When working with simple applications, developers choose static websites for their simplicity, and they avoid using complicated methods like scripting. These applications that do not require constant updates tend to utilize only two languages: HTML and CSS. Using only HTML and CSS results in a simple website that allows customization for each webpage but lacks dynamic capabilities.

Real Time/ Cloud-based

Websites that change as information updates typically require both the backend that includes a server, an application, and a database and the frontend that includes web design and front end web development [1]. A popular method of linking the backend and the frontend is Flask, a microframework for Python that runs an application that relies on a HTML file for the frontend and a python file for data handling and other responsibilities. Flask creates multiple URL paths for the website, handles HTTP methods, renders HTML templates, and sends requests [2]. Flask's capabilities add the dynamic aspect to a website as the python script runs

the application, requests new information, and updates the HTML file with the retrieved information.

The cost of the website depends “on several factors such as the number of Amazon EC2 instances needed to handle your web site traffic, the bandwidth consumed by your application, and which database or storage options your application uses [3].” The files (Python, HTML, CSS) are hosted on an Amazon EC2 instance while the data is streamed through Amazon S3 storage. For the EC2 instance, the cost can vary from $0.0058 per hour to $3.2 per hour while the S3 storage adds an additional $0.023 per GB [4] [5].

Technology Gathering Real Time Data

A single board computer like the Raspberry Pi Zero W has an assortment of functionalities, but one in particular runs a Node.js webserver from the device and allows a user to control the Pi’s inputs and outputs [6]. The Pi takes data readings from an input and uploads it to the webserver. A Python script using Flask requests the data from the webserver and displays the information live on a website.

The two popular single board computers are C.H.I.P. and Pi Zero W. Both have Wi-Fi and other similar features, but the main difference is the price: the cost of the CHIP Pro is $16 while the Pi Zero W is $10.

Works Cited

[1] J. Long, "I Don’t Speak Your Language: Frontend vs. Backend," 25 2012 2012. [Online]. Available: http://blog.teamtreehouse.com/i-dont-speak-your-language-frontend-vs-backend. [Accessed 24 October 2017].

[2] A. Ronacher, "Quickstart," [Online]. Available: http://flask.pocoo.org/docs/0.12/quickstart/. [Accessed 24 October 2017].

[3] Amazon, "AWS Elastic Beanstalk Pricing," [Online]. Available: https://aws.amazon.com/elasticbeanstalk/pricing/ . [Accessed 24 October 2017].

[4] Amazon, "Amazon EC2 Pricing," [Online]. Available: https://aws.amazon.com/ec2/pricing/on-demand/. [Accessed 24 October 2017].

[5] Amazon, "Amazon S3 Pricing," [Online]. Available: https://aws.amazon.com/s3/pricing/. [Accessed 24 October 2017].

[6] Raspberry Pi Tutorials, "How to setup a Raspberry Pi Node.js Webserver and control GPIOs," [Online]. Available: https://tutorials-raspberrypi.com/setup-raspberry-pi-node-js-webserver-control-gpios/ . [Accessed 24 October 2017].