[[1]](#footnote-1)

HTML Educational Node.js System (HENS)

An applied system for web development

Brian Carter*, Chippewa Software Technology*

*Abstract*— In this paper, a very compelling proposition is presented, a portable system for software development. The focus is on HTML, JavaScript, CSS, Node.js, and SQLite. The system provides all the applications required for students to start developing web applications. The HTML Educational Node.js System (HENS) does not require installation or admin access allowing for portability. This environment is built using open source applications.

*Index Terms*—education, development environment, portal applications, website, web service, html, javascript, node.js

# Introduction

Web development has become one of the most in-demand skills. To meet this need, the addition and retention of students in this field is critical. Web development has a steep learning curve and many obstacles. One obstacle is setting up a development environment. Many students get frustrated before they even write their first web application due to the complexities in setting up their development environment.

This paper’s position is to develop a system that allows students to focus on web development and to allow initial classes of a web development course to focus on the software development aspects instead of environment setup. Typically, a majority of a course is spent getting the onsite and students’ environments set up to begin the instructions on software coding. This is partially due to the barriers in the environments: lack of admin rights to install software, blocked ports, limited base installs, and the steep learning curve and installation requirements of virtual machines. Another barrier is that students move from class to class, from home to public computers, to common areas for working on projects; unable to bring with them the lab machine that they configured. Another barrier, after the time consuming and frustrating setup tasks are completed, a simple mistake can require the installation process to be redone.

HTML Educational Node.js System (HENS) aims to remove some of this frustration. The strategy is to use a portable drive (portable hard drive, USB, cloud drive) as the development environment. The portable drive would have everything the student requires to develop a web-based application. This

includes tools for HTML, JavaScript, and CSS for front end development work. An important part of teaching web development is the backend, or server coding. HENS uses Node.js as the server side development platform and SQLite as the database environment.

In our labs, it has been found that the best approach is to have the instructor create the base build. This allows the students to simply copy and paste the environment to their portable drives. As the course continues and students get familiar with the tools, it is a good exercise to have the students build HENS. This knowledge will position the students to add additional tools, packages, tools, and data stores.

# Installation

The HENS foundation uses PortableApps [1]. This provides a base that is portable and well supported by the community. PortableApps allows for installation without admin rights and targets portable drives. This provides a base that enables students to carry all their development applications on a portable drive and use it on any Personal Computer (PC). Their settings and configurations will be retained in HENS; allowing students to switch PCs and continue where they left off. Work, home, school, visiting family and friends, even while traveling, everything stays on the portable drive. Every PC becomes a student’s development PC. Working in distributed work environments is becoming the norm for both professionals and students. Educators must take action to prepare students to be productive in such environments [2]. HENS is built to support this need.

Why use the Microsoft Operating System (OS)? Microsoft is still the dominating desktop in homes, schools, and corporations. The NetMarketShare report [3] shows the Microsoft OS dominates well over 70% of desktops. To reach the majority of students, HENS targets the Microsoft OS; the instructions found in this paper can be used on other operations systems with some customization.

The first step is to install PortableApps on the portable drive. Download the application and follow the instructions for installing to the root of a portable drive [1]. Once installed, run the PortableApps platform found on the portable drive (PortableApps.com/PortableAppsPlatform.exe) as shown in Figure 1.

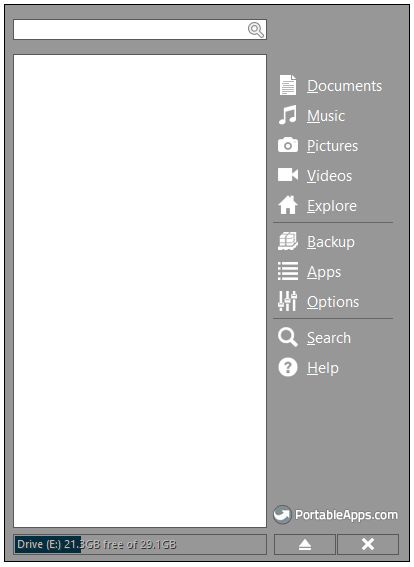


Fig. 1. HENS base PortableApps platform.

With the platform installed, use the Apps menu item to install the tools shown in Table 1. Selecting Apps menu item, the Portable Apps Directory allows for installation of supported applications (figure 2).

TABLE I

HENS base apps to install

1. Notepad++
2. SQLite Database Browser
3. Google Chrome

Notepad++ is the tool for creating and updating code (IDE: Interactive Development Environment). The SQLite Database Browser is the graphical UI tool for working with SQLite databases. Google Chrome is a web browser with built in development tools.

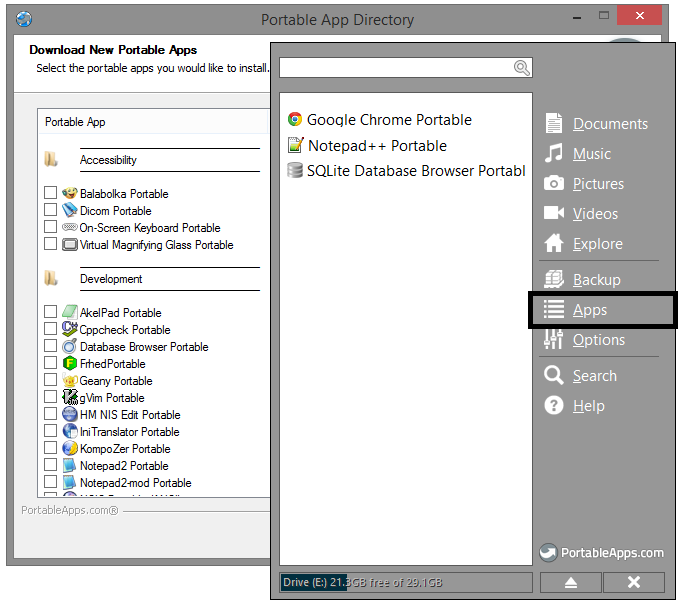


Fig. 2. Using PortableApps platform to install Apps.

# Server side development

At this point, HENS has all the tools required for creating HTML, CSS, and JavaScript (often called frontend development). The majority of web applications require server side processing and access to a data store (often called backend or server side development). This is where a console application is added to HENS to provide a server environment for backend development.

A console application (console) will be set up and configured as the server. A console, similar to the Linux shell, has the ability to run Node.js [4] to server HTML pages with backend logic. A command or script will be needed to start the working environment and set the appropriate path variables. In Windows, the full path must be given to run a program or the programs path must be in the path variable.

Navigate to the portable drive. Create a directory called “serverportable” in the root with a folder called “bin”. The folder “serverportable\bin” will hold the executables required for HENS backend development.

Using HENS Notepad++, create a new file “startserver.bat” (batch file) and save it to the portable drive folder “\serverportable”. The batch file is needed to start the working environment and set the appropriate path variables. It will provide a self-contained and portable script that sets the paths for all server side tools. While this can be done by running each command individually, it is easier for the student to run the batch file instead of an error prone process of running each command individually. The complete startserver.bat can be found in the appendix.

# Node.js

Node.js is a platform built on Chrome’s JavaScript runtime for easily building fast, scalable network applications [4]. Node.js is an open source, cross-platform runtime environment for server-side applications. Node.js applications are written in JavaScript and supports the concept of packages to interact with data stores and other server resources. The Node.js runtime is supported on Microsoft Windows, Linux, and OS X. HENS uses Node.js for its backend development and web server.

On the portable drive, create a folder for installing node: “serverportable\bin\node”. Download node.exe from Nodejs.org (<http://nodejs.org/dist/latest/>); installation is as simple as copy and paste. Using Notepad++, update the batch file to contain:

: @echo off

: set a reference to server dir

set SERVER\_DIR=%~p0

: change to server dir

cd /D SERVER\_DIR

: add node.js dir to path

set NODEDIR=%SERVER\_DIR%bin\node

set PATH=%NODEDIR%;%PATH%

set NODE\_PATH=%NODEDIR%\node\_modules\;%NODEDIR%\node\_modules\npm\node\_modules;%NODEDIR%\node\_modules\npm;%NODEDIR%\node\_modules\sqlite3;%NODEDIR%\node\_modules\nodemon;

The batch file updates set the SERVER\_DIR variable to the server directory. The script creates a NODEDIR variable for the path where node is installed. The variables will be used as a base for other script set commands.

While you are installing node, go ahead and install the node package manager [5]. Download the latest version (<http://nodejs.org/dist/npm>) and unzip it to the node subdirectory; the portable drive should now have the following folder structures. Notice that node\_modules and npm are both in the set path statements of the batch file.

“ServerPortable\bin\node\node\_modules\npm”

For your code, create the “code” folder in the root of the portable drive. This is where all our code will reside. Update the batch file to include starting the command console to run DOS command. Add this line to the end:

start cmd.exe /k "cd ..\code"

Now run the batch file (double click). A console window will appear and it will be in the code directory as shown in figure 3 (a 2nd console window will be open – showing the execution of the script. This can be closed). To test your installation, run the following commands in the console window:

node –version

npm --version

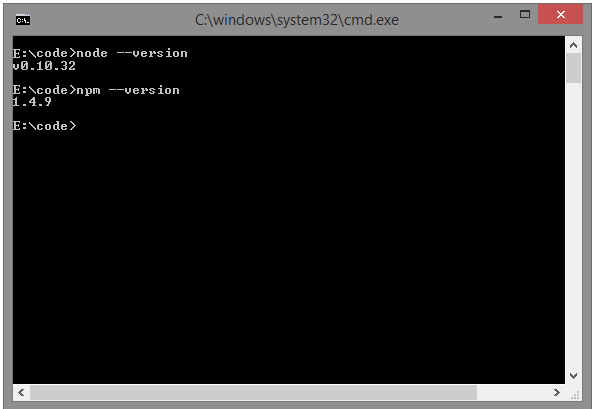


Fig. 3. HENS testing installation of node and npm, shows versions.

If everything is installed correctly, your results should be similar to figure 3 (version numbers may be different or newer).

Let’s test a simple Node.js application on HENS. In Notepad++, create a new file and save as “app.js” in the code folder. Add the following to the app.js file:

var http = require('http');

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/plain'});

res.end('Hello World\n');

}).listen(8080, '127.0.0.1');

console.log('Server running at http://127.0.0.1:8080/');

In the console, you should still be in the code folder. Test the application by running the following at the command prompt:

node app.js

Run Chrome from the PortableApps platform and navigate to the web page: <http://127.0.0.1:8080/>. You should see “Hello World” as shown in figure 4.



Fig. 4. HENS app.js test.

With npm, additional packages can be installed based on your class requirements. For example, nodemon is a great tool that will watch for files in the directory to change and will restart the application automatically, otherwise node must be restarted after each update. In your console run the following commands to install nodemon:

npm install -g nodemon

For details on additional Node.js packages, see npm’s project site [5]. Using nodemon is outside the scope of this paper, see the project site for additional details: https://github.com/remy/ nodemon#nodemon.

# database development

Web applications typically require a data store. For HENS, SQLite is used as the data store. SQLite works well with HENS because it is not only headless, but a simple relational database that can be used by Node.js. SQLite is easy to setup, connect to, query, and is well supported in the open source community. It also supports portability by storing everything in a single file [6].

There are many Node.js modules out there that can be used to manage SQLite databases. They off work off the principle of providing a simple API, buffer support, serialization, and performance. For HENS, the node module sqlite3 is used to communicate with the database file.

In the console, install SQLite by running the following command. This command will install sqlite3 in the “ServerPortable\bin\node\node\_modules\sqlite3” directory. Using npm makes this an easy tasks:

npm -g install sqlite3

SQLite also provides a command line shell (http://www.sqlite.org/download.html). This provides HENS access to SQLite using command-line commands. Download the percompiled binaries for Windows and unzip the “sqlite3.exe” to the portable drive folder: “ServerPortable\bin\ sqlite\bin”. The SQLite paths must be added to the batch file:

: add sqlite directory and path

set PATH=%SERVER\_DIR%bin\sqlite\bin;%PATH%

To test the installation of SQLite, run the version command. To test the creation of a database, run the following commands. The results are shown in figure 5.

sqlite3 -- version

sqlite3 ex1

sqlite> create table tbl1(one varchar(10), two smallint);

sqlite> insert into tbl1 values('hello!',10);

sqlite> insert into tbl1 values('goodbye', 20);

sqlite> select \* from tbl1;

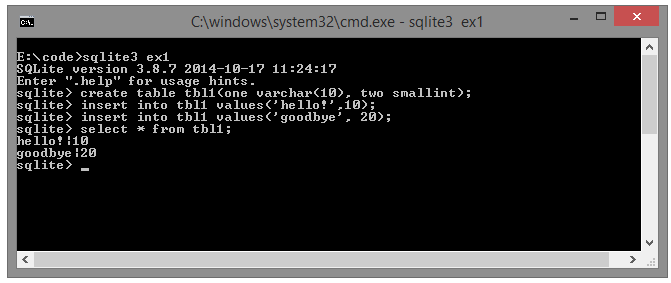


Fig. 5. HENS SQLite command line shell results.

The commands create a database file called ex1. In Section II, the SQLite Database Browser was installed. Click on the menu item in the PortableApps Platform. As shown in figure 6, open the ex1 database to manage the database in the application. Often, students find the graphical user interface easier to understand than command line SQL statements.

The concepts behind relational databases and SQLite are beyond the scope of this paper. More details can be found [7].

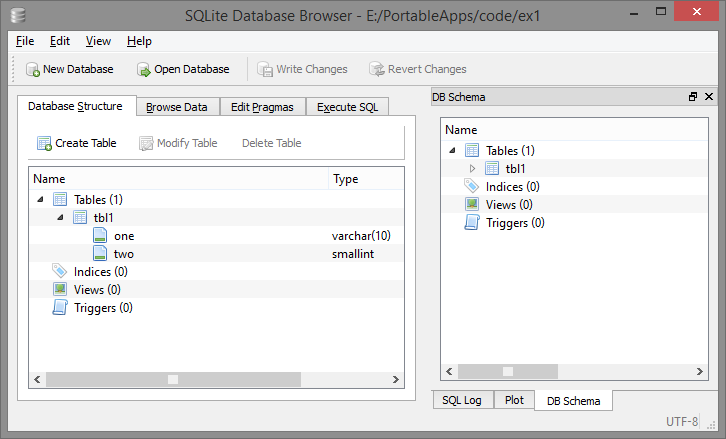


Fig. 6. SQLite Database Browser.

# version control

Version control is a system that will track all code changes. Git is a distributed version control that is popular for open source projects. Even in this paper, our batch file was created and updated several times. Using Git as the version control, all changes can be tracked.

Many open source projects use GitHub[8]. This is a free Git repository in the cloud. HENS requires Git to be installed to allow students to push and pull file changes from GitHub. Download the latest executable from git for Windows (http://msysgit.github.io/). Running the executable file will install git to the “server” folder. Copy the “server\bin\git” folder to the portable drive “serverportable\bin\git”.

With Git installed, the folders need to be added to the batch file. Add the following lines before the start cmd line:

: add the git directory and the minimal UNIX to the path

set PATH=%SERVER\_DIR%bin\git\cmd;%PATH%

set PATH=%SERVER\_DIR%bin\git\bin;%PATH%

To test if the Git files were moved correctly and the paths added to the batch file, run the following command in console. The result should show the version:

git –version

Using Git and GitHub is outside the scope of this paper. More details can be found [9].

# Backup and recovery

The tasks presented in this paper installed HENS to the root of the portable drive. Backing up HENS is a simple task of copying all the files to another location. For delivery to students, backing up to a zipped file is recommended. This provides a good starting point. Have the students copy the zip file to the root of their portable drive. Then unzip the files.

During class, many changes can be made. After updates, have the students create a zipped version of their own files. This gets them into the practice of backing up. To test the backup, have the students use another portable drive or create a “restored” folder. Unzip a version of their HENS and test the backup.

Delivery of class code examples should use GitHub. HENS has git installed and students can pull down class code examples to the code folder with the following simple “git clone” command.

# conclusion

In this paper, HENS was proposed as an environment that provides a portable education system for software development. Creating HENS is involved and requires many steps. The benefit is the creation of a system that can be copied to students’ portable drives that contains the tools required to develop web applications. HENS provides an environment that is accessible to beginning students while providing a professional portable environment for seasoned developers. HENS and items in this paper can be downloaded from GitHub [10].

# Appendix

Complete source code for startserver.bat.

: @echo off

: set a reference to server dir

set SERVER\_DIR=%~p0

cd /D SERVER\_DIR

: add the git directory and the minimal UNIX to the path

set PATH=%SERVER\_DIR%bin\git\cmd;%PATH%

set PATH=%SERVER\_DIR%bin\git\bin;%PATH%

: add sqlite directory and path

set PATH=%SERVER\_DIR%bin\sqlite\bin;%PATH%

: add node to the path and set NODE\_PATH (where node searchs for modules)

set NODEDIR=%SERVER\_DIR%bin\node

set PATH=%NODEDIR%;%PATH%

set NODE\_PATH=%NODEDIR%\node\_modules\;%NODEDIR%\node\_modules\npm\node\_modules;%NODEDIR%\node\_modules\npm;%NODEDIR%\node\_modules\sqlite3;%NODEDIR%\node\_modules\nodemon;

start cmd.exe /k "cd ..\code"

# References

1. PortableApps.com - Portable software for USB, portable and cloud drives. (n.d.). Retrieved from <http://portableapps.com>.
2. Meneely, A., & Williams, L. (2009). On preparing students for distributed software development with a synchronous, collaborative development platform. *ACM SIGCSE Bulletin,* 529-529.
3. Operating system market share. (n.d.). Retrieved from <http://www.netmarketshare.com/operating-system-market-share.aspx>, November 1, 2014.
4. Node.js. (n.d.). Retrieved from http://nodejs.org, November 1, 2014.
5. Node Packaged Modules. Retrieved from <http://nodejs.org/dist/npm>, November 1, 2014.
6. SQLite. Retrieved from http://www.sqlite.org, November 1, 2014.
7. Kreibich, J. (2010). *Using SQLite*. Sebastopol, CA: O'Reilly.
8. Build software better, together. (n.d.). Retrieved from <http://www.github.com>, November 1, 2014.
9. Loeliger, J. (2009). *Version control with Git*. Beijing: O'Reilly.
10. HENS. Retrieved from https://github.com/ChipSoftTech/HENS.

1. This work was supported in part by Sullivan University and Chippewa Software Technology.

   Brian Carter is with Sullivan University, College of Information and Computer Technology, Louisville, KY 40205 USA (e-mail: [bscarter@sullivan.edu](mailto:bscarter@sullivan.edu)) and is with Chippewa Software Technology, La Grange, KY 40031 USA (e-mail: [briancarter@chipsofttech.com](mailto:briancarter@chipsofttech.com)). [↑](#footnote-ref-1)