

## 1. Testing with Node using Mocha

Write unit tests for problems 1-5 using Mocha/Chai. The code for the tests should be written in the file *node\_tests / tests / app.test.js* and should test the functions already created in the file *node\_tests / app.js* (the functions are the same as in the file above, only the app.js file has been slightly edited to accomodate Node).

To run the tests with Node, do the following steps:

- a. Install Node.js if you haven't already:  
<https://nodejs.org/en/download/>
- b. Open the directory (folder) *node\_tests* in the terminal and use this command:

```
npm init
```

(npm stands for Node Package Manager)

You will be asked to set the package name, version, etc. The only thing you have to set is the test command, which should be:

```
./node_modules/.bin/mocha ./test/app.test.js
```

This will create the file *package.json*, which npm uses to install dependencies for you project.

- c. Almost there! Now install and add Mocha and Chai to your dependencies with:

```
npm i --save-dev mocha chai
```

- d. Now you can run your tests with:

```
npm test
```

## 2. Testing in the browser using Mocha

Write unit tests for problems 1-5 using Mocha/Chai. The code for the tests should be written in the file *browser\_tests / test / app.test.js* and should test the functions already written in the file *browser\_tests / app.js* . You can see if the tests passed or failed by opening the file *browser\_tests / d4.html* in a browser of your choice.