

Objective

Deploy our lab9 application using Heroku.

Step 1

Signup with Heroku and create a new app:

Create New App


App name

bmrlab10


✓

bmrlab10 is available

Choose a region

 United States

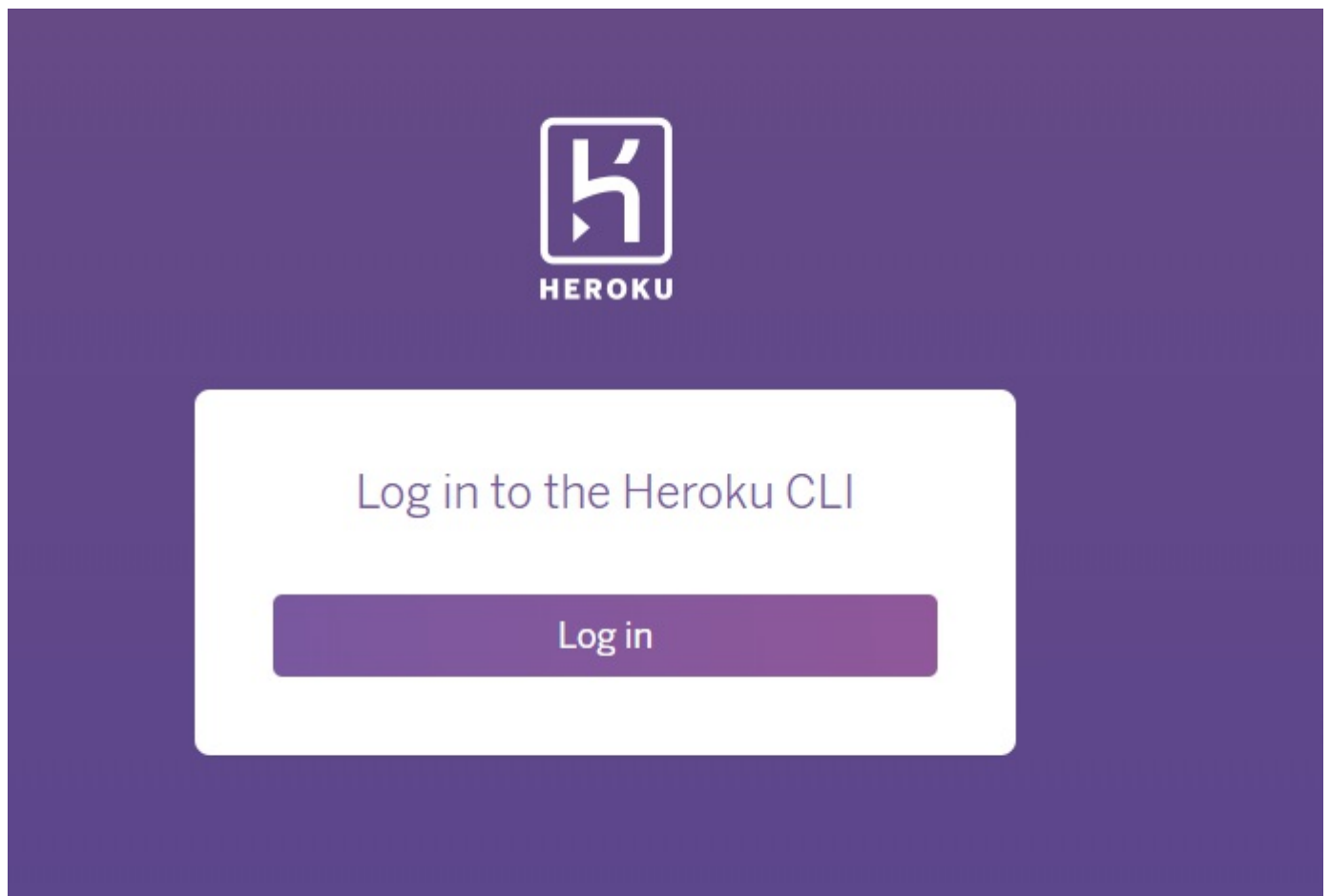
⌵

 Add to pipeline...

Create app

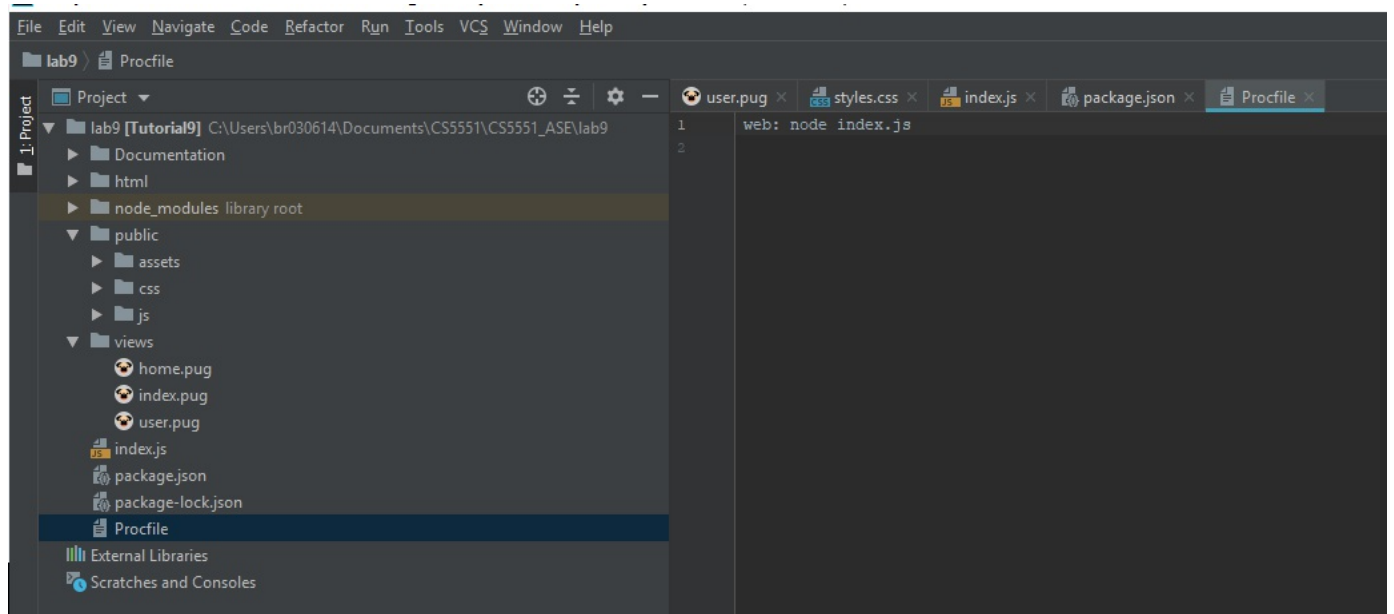
Step 2

Download Heroku CLI and login through the CLI:



Step 3

Add Procfile to existing project:



Step 4

Determine the current version of node and npm installed locally:

```
BR030614@W1644226 MINGW64 ~/Documents/CS5551/CS5551_ASE/lab9 (master)
$ node -v
v10.15.0

BR030614@W1644226 MINGW64 ~/Documents/CS5551/CS5551_ASE/lab9 (master)
$ npm -v
6.9.0
```

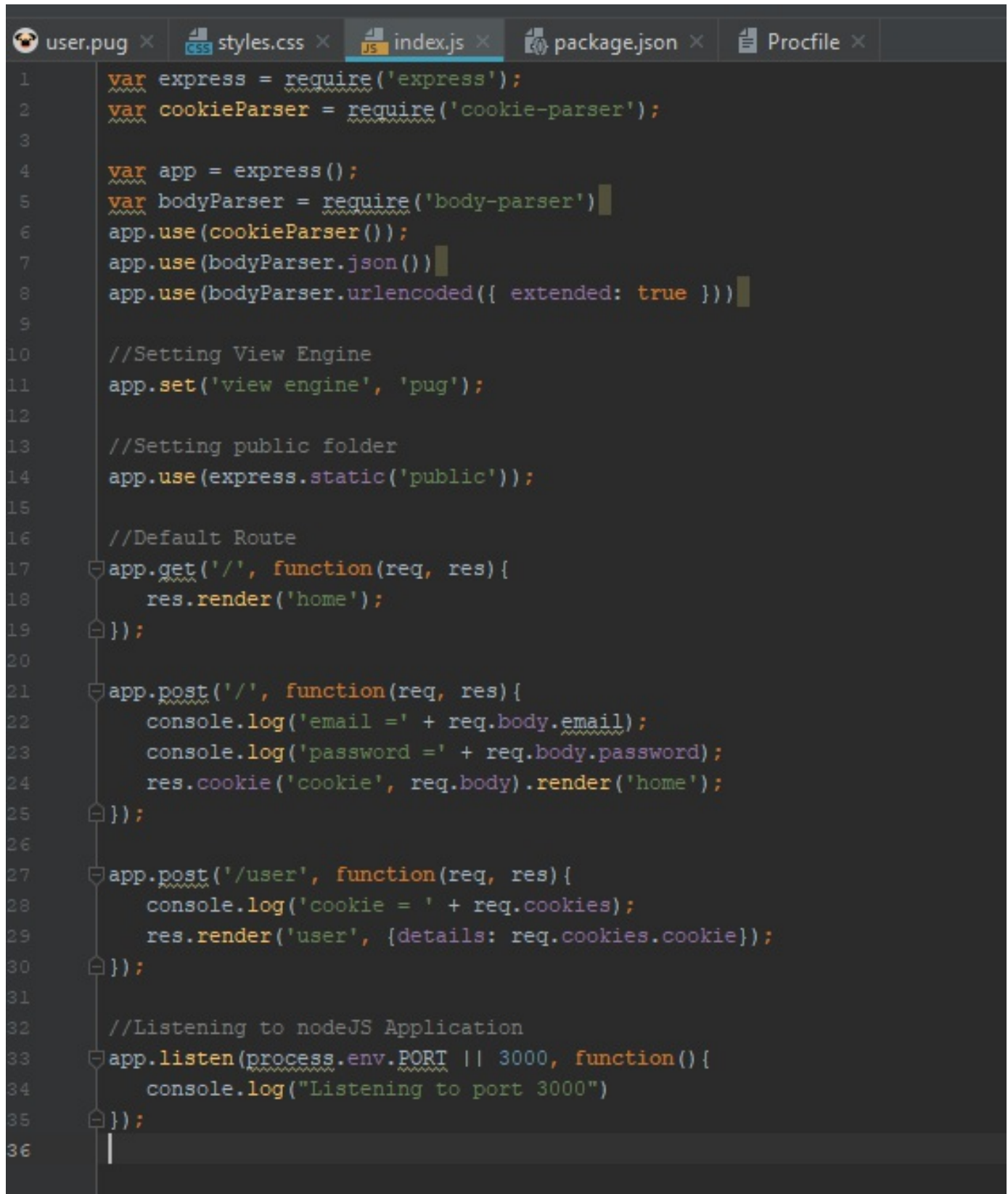
Step 5

Update package.json file with details from step 4:

```
{
  "name": "tutorial9",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.18.3",
    "bootstrap": "^4.3.1",
    "cookie-parser": "^1.4.4",
    "express": "^4.16.4",
    "pug-cli": "^1.0.0-alpha6"
  },
  "engines": {
    "node": "10.15.0",
    "npm": "6.9.0"
  }
}
```

Step 6

Update index.js file to scan for a port instead of being hardcoded to port 3000:



```
1  var express = require('express');
2  var cookieParser = require('cookie-parser');
3
4  var app = express();
5  var bodyParser = require('body-parser');
6  app.use(cookieParser());
7  app.use(bodyParser.json());
8  app.use(bodyParser.urlencoded({ extended: true }));
9
10 //Setting View Engine
11 app.set('view engine', 'pug');
12
13 //Setting public folder
14 app.use(express.static('public'));
15
16 //Default Route
17 app.get('/', function(req, res){
18   res.render('home');
19 });
20
21 app.post('/', function(req, res){
22   console.log('email =' + req.body.email);
23   console.log('password =' + req.body.password);
24   res.cookie('cookie', req.body).render('home');
25 });
26
27 app.post('/user', function(req, res){
28   console.log('cookie = ' + req.cookies);
29   res.render('user', {details: req.cookies.cookie});
30 });
31
32 //Listening to nodeJS Application
33 app.listen(process.env.PORT || 3000, function(){
34   console.log("Listening to port 3000")
35 });
36
```

Step 7

Initialize git repo in project folder:

```
BR030614@W1644226 MINGW64 ~/Documents/CS5551/CS5551_ASE/lab9 (master)
$ git add .
warning: LF will be replaced by CRLF in .idea/workspace.xml.
The file will have its original line endings in your working directory
```

Step 8

Commit project to newly created git repo:

```
BR030614@W1644226 MINGW64 ~/Documents/CS5551/CS5551_ASE/lab9 (master)
$ git commit -am "lab 10"
[master 77795af] lab 10
1 file changed, 7 insertions(+), 9 deletions(-)
```

Step 9

Push commit to heroku master:

```
BR030614@W1644226 MINGW64 ~/Documents/CS5551/CS5551_ASE/lab9 (master)
$ git push heroku master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 501 bytes | 501.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
```

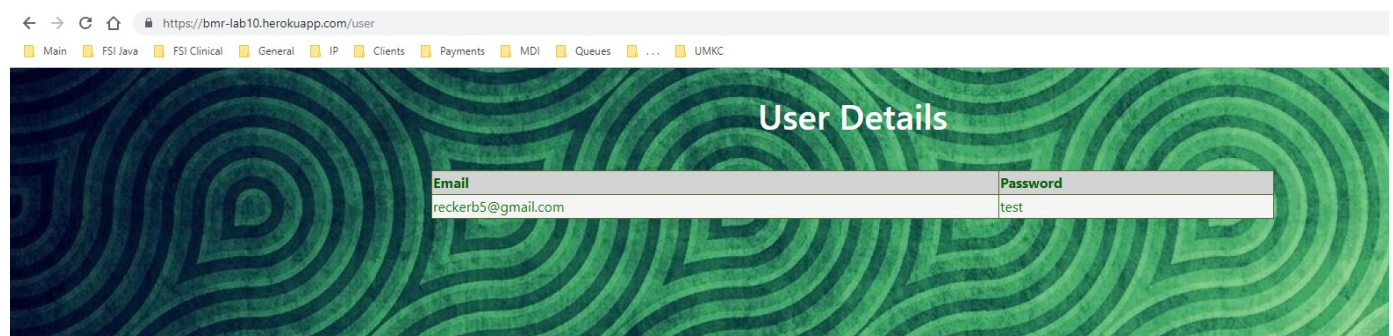
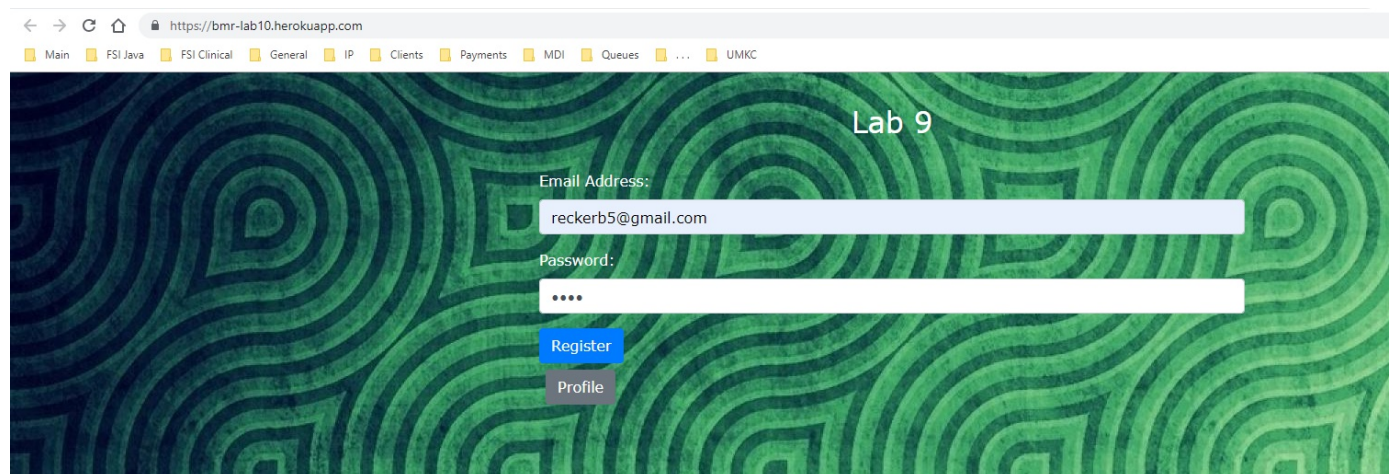

Step 10

Determine project's Heroku URL:

```
remote:
remote: -----> Compressing...
remote:           Done: 30.1M
remote: -----> Launching...
remote:           Released v8
remote:           https://bmr-lab10.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/bmr-lab10.git
    7de4c93..77795af  master -> master
```

Step 11

Navigate to URL and test application:



HEROKU URL

<https://bmr-lab10.herokuapp.com/>

Outcome

Deploying my project to Heroku was very straight forward and simple. I followed the steps above and only had 1 issue. I initially named the procfile with a lowercase p. Using '*heroku logs --tail*' gave me an error stating this should be renamed with an uppercase P (Procfile). Once I updated this and pushed the changes to the master my application was working as expected.