



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Puzzle Generator: Technical installation manual

Team Prometheus

16 September 2020

Clients: Mark Anthony and Francois

Name	Student Number	Email
Yuval Langa	u18174142	u18174142@tuks.co.za
Tsholofelo Gomba	u17391718	u17391718@tuks.co.za
Tapiwa Mazibuko	u18203541	u18203541@tuks.co.za
Charlotte Jacobs	u15165622	u15165622@tuks.co.za
Jaynill Gopal	u15175295	u15175295@tuks.co.za

Contents

1	Server setup - Backend	3
1.1	Prerequisites	3
1.2	Installation on localhost	3
2	Deploying to Heroku	5
2.1	Setting up your database	15
2.2	Customizing your server	16
2.2.1	Changing server's default port	16
2.2.2	Adding your web pages to be deployed	17
3	Angular Usage - Front-end	17
3.1	Development server	17
3.2	Code scaffolding	17
3.3	Build	17
3.4	Running unit tests	17
3.5	Further help	17
3.6	Angular File Structure	18
3.7	Main Folder	18
3.8	src folder	18
3.9	assets folder	18
3.10	app folder	18
3.11	models folder	18
3.12	services folder	18
3.13	pages folder	18

1 Server setup - Backend

1.1 Prerequisites

Ensure that you have the following installed on your system.

1. NodeJS version 12.0+
2. npm version 4.0+
3. Angular CLI via NPM
4. PostgreSQL

1.2 Installation on localhost

In order to use the server and load files to be hosted on the website the following needs to be done:

1. Open terminal to the directory which you had cloned/ downloaded the source code from the Puzzle-Generator GIT
2. Enter the following command to install dependencies: **npm install**

```
C:\Users\... \GIT\Puzzle-Generator>npm install
npm WARN deprecated @angular/http@7.2.16: Package no longer supported. Use @angular/common instead, see https://angular.io/guide/migration
npm WARN deprecated core-js@2.6.11: core-js@<3 is no longer maintained and not recommended for usage due to the number of d...
```

3. Once the dependencies have been downloaded and installed, a post installation task will automatically be run in order to build the relevant Angular files which reside in the subdirectory **/src** and output the built application in the directory **/dist**.

```
> ng build --output-path dist
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.
chunk {polyfills} polyfills-es2015.js, polyfills-es2015.js.map (polyfills) 141 kB [initial] [rendered]
chunk {polyfills-es5} polyfills-es5.js, polyfills-es5.js.map (polyfills-es5) 752 kB [initial] [rendered]
chunk {main} main-es2015.js, main-es2015.js.map (main) 156 kB [initial] [rendered]
chunk {main} main-es5.js, main-es5.js.map (main) 170 kB [initial] [rendered]
chunk {runtime} runtime-es2015.js, runtime-es2015.js.map (runtime) 6.16 kB [entry] [rendered]
chunk {runtime} runtime-es5.js, runtime-es5.js.map (runtime) 6.16 kB [entry] [rendered]
chunk {styles} styles-es2015.js, styles-es2015.js.map (styles) 156 kB [initial] [rendered]
chunk {styles} styles-es5.js, styles-es5.js.map (styles) 158 kB [initial] [rendered]
chunk {vendor} vendor-es2015.js, vendor-es2015.js.map (vendor) 5.07 MB [initial] [rendered]
chunk {vendor} vendor-es5.js, vendor-es5.js.map (vendor) 5.92 MB [initial] [rendered]
Date: 2020-07-23T20:04:36.172Z - Hash: cf9cc37ce258023eb80a - Time: 60058ms
```

4. If a directory **/dist** is not created after calling the command **npm install**, you may use the command **npm postinstall** which will run the command **ng build --output=dist**.

5. In order for the system to run on your localhost, please indicate the following environment variables in order to connect to your postgres database:

(a) In order to set environment variables in Windows:

- From the desktop, right click the Computer icon.
 - Choose Properties from the context menu.
 - Click the Advanced system settings link.
 - Click Environment Variables
 - Under user variables click on new and enter the environment variables listed below
- dbDatabase: The database which you will be using
 - dbHost: The link to the database
 - dbpass: The database's password
 - dbUsername: The username which you login to the database
 - emailPassword: The password for the email which your system will be sending emails to users
 - emailUname: The email address for the email which your system will be sending emails to users

User environment variables:

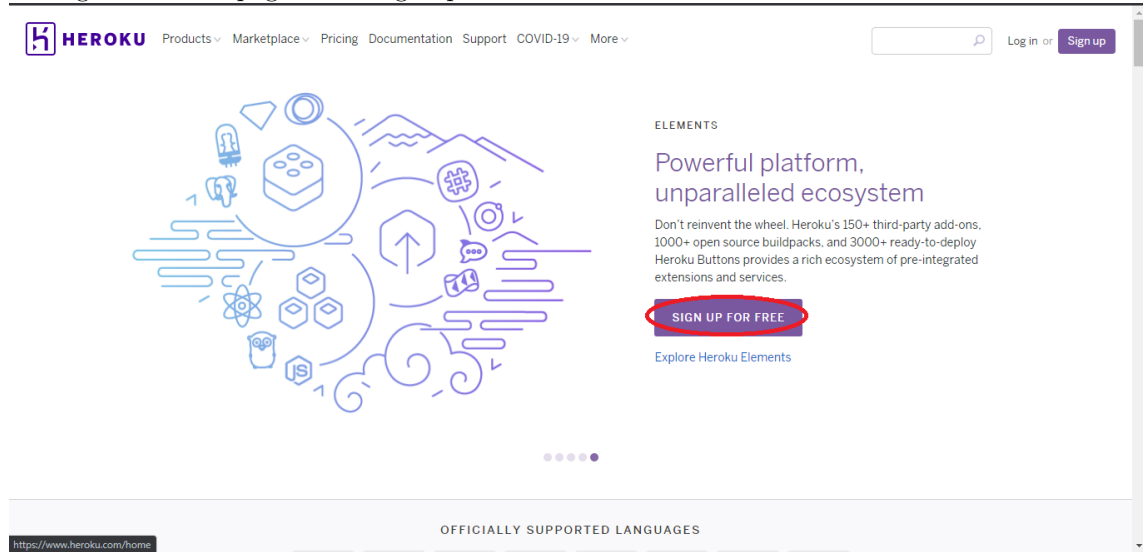
Name	Value
dbDatabase	d...
dbpass	aC...57...
dbUsername	y...n
dbHost	ec2-...amazonaws.co

6. You are now ready to run the server.
7. To run the server enter the following command into your terminal in the root folder of your app: **npm start**.
8. You are now ready to access your server at **localhost:3200** in your browser.


2 Deploying to Heroku

In order to deploy to Heroku, the following needs to be done:


- Go to the Heroku website at the following link <https://www.heroku.com/>
- Naviagte the home page to the sign up link.




- Enter your details in the sign-up sheet

 **Free account**

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

 **Your app platform**

A platform for apps, with app management & instant scaling, for development and production.

 **Deploy now**

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name *

Team

Last name *

Prometheus

Email address *

team.prometheus@willowmoore.co.za

Company name

University of Pretoria

Role *


Student


Country *

South Africa

Primary development language *

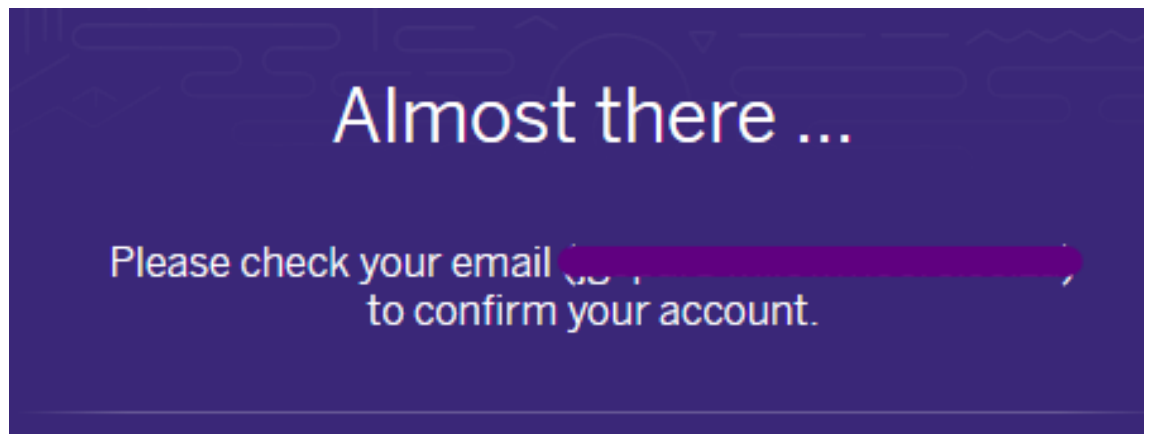
Node.js

 I'm not a robot


reCAPTCHA
[Privacy](#) - [Terms](#)

CREATE FREE ACCOUNT

- Once you have signed up, you will be taken to the signup confirmation page



- Go to your inbox and click on the link sent to you by Heroku



Thanks for signing up with Heroku! You must follow this link to activate your account:

<https://id.heroku.com/account/accept/22222222-22222222-22222222-22222222>

Have fun, and don't hesitate to contact us with your feedback.

The Heroku Team
<https://heroku.com>

- Upon clicking the confirmation link, you will be required to create your password

Set your password

Create your password and log in to your Heroku account.

Create a new password *

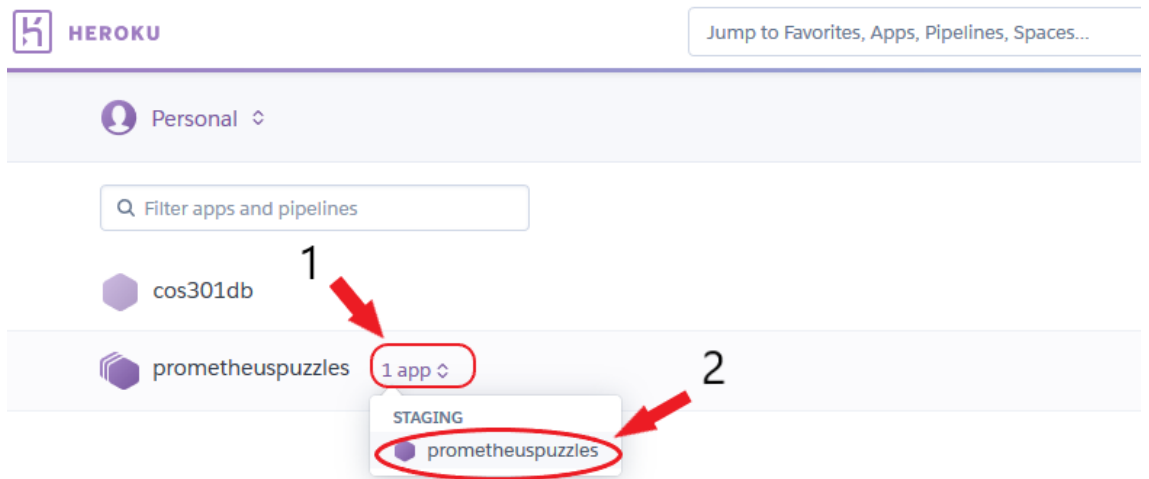
Password confirmation *

Password requirements

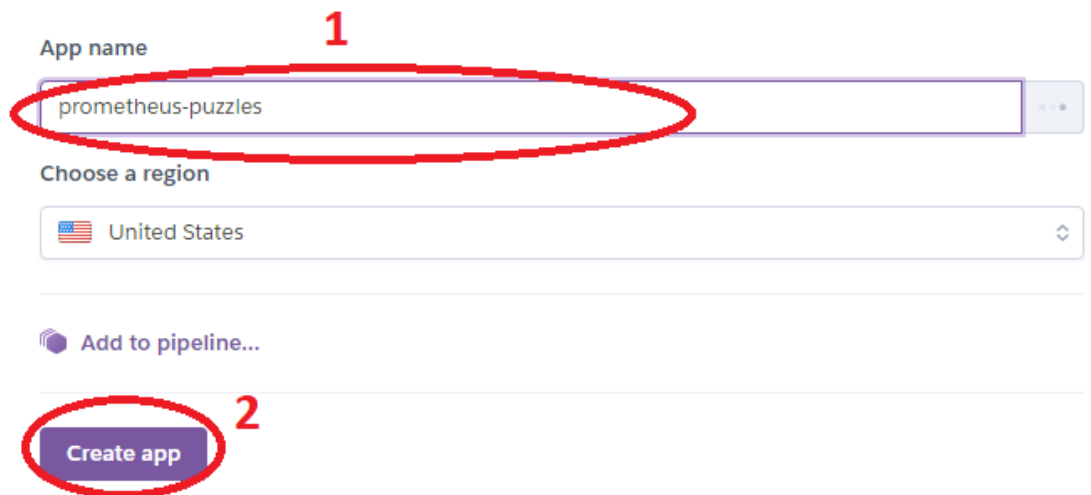
- ✓ Must be a minimum of 8 characters.
- ✓ Must contain letters, numbers, and symbols.
- ✓ Passwords must match.

SET PASSWORD AND LOG IN

- After clicking on "Set Password and Login", you will be taken to your Heroku Dashboard. From this page click on "Create new app"



- Once clicking on this link, you will be asked to enter your app's name.



- After which, you will need to click on "Deploy" and then, select GitHub as your deployment method, then select "Connect to GitHub"

Personal > prometheuspuzzles > prometheus-puzzles ☆ Open app

Overview Resources **Deploy** Metrics Activity Access Settings

Connected to a pipeline **1**

Assigned to `staging` in prometheuspuzzles

Manage this pipeline and this app's stage on the [pipeline overview](#)

Review apps are available on the [pipeline overview](#)

Deployment method

Heroku Git Use Heroku CLI **GitHub** Connect to GitHub Container Registry Use Heroku CLI **2**

Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

Deployment method

Heroku Git Use Heroku CLI GitHub Connect to GitHub Container Registry Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

View your code diffs on GitHub

Connect your app to a GitHub repository to see commit diffs in the activity log.

Deploy changes with GitHub

Connecting to a repository will allow you to deploy a branch to your app.

Automatic deploys from GitHub

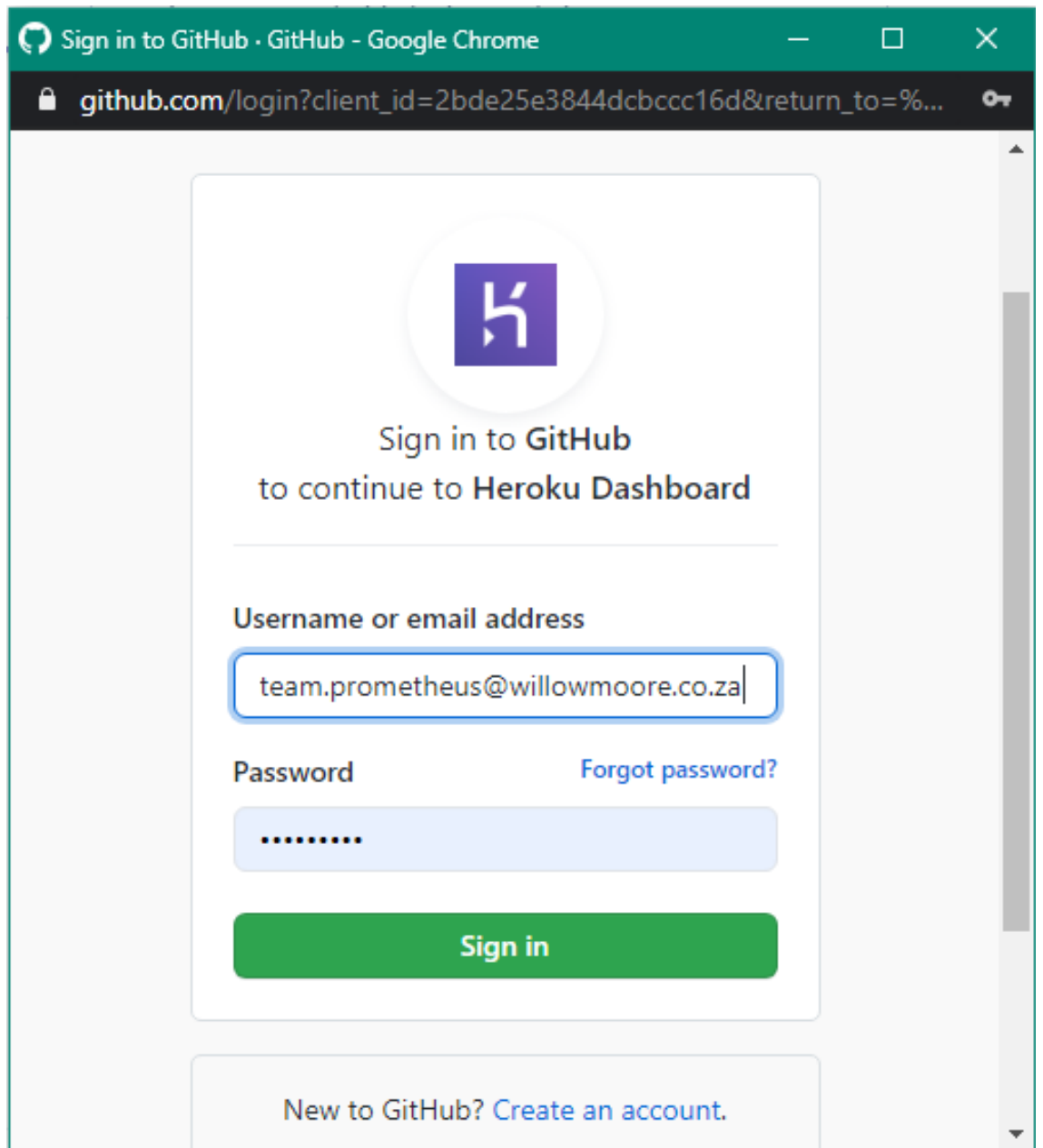
Select a branch to deploy automatically whenever it is pushed to.

Create review apps in pipelines

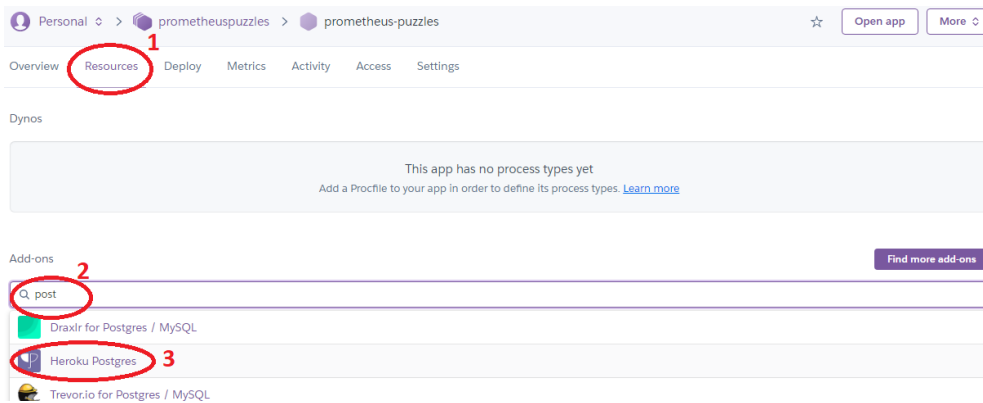
Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more](#).

Connect to GitHub

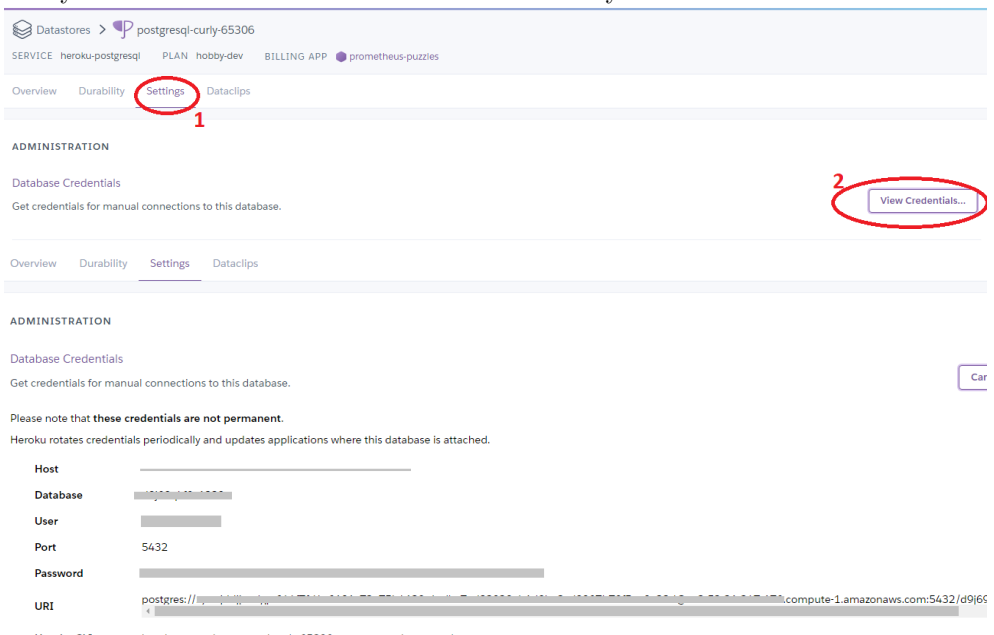
- You will be asked to authorize Heroku with GitHub, a popup will show asking you to enter your GitHub credentials



- Once connected to your GitHub page, select the "Resources" tab on the navigation bar. Then select the search bar under addons and type "Postgres" and select "Heroku Postgres"

















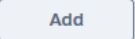
- Select the "Heroku Postgres" link after which a page will show up with the overview of the database. Select the "Settings" link and then view credentials for the database. After which the system will reveal the necessary credentials needed to connect the website to your database.





- Return to your dashboard, by going to `dashboard.heroku.com`, and selecting **prometheuspuzzles**
- Select settings and navigate to the "Reveal Config Vars" button and select the button. Enter your credentials in the relevant details Namekey:
 - dbDatabase: The database which you will be using
 - dbHost: The link to the database

- dbpass: The database’s password
- dbUsername: The username which you login to the database
- emailPassword: The password for the email which your system will be sending emails to users
- emailUname: The email address for the email which your system will be sending emails to users


Config Vars
Hide Config Vars

DATABASE_URL	postgres://	 
dbDatabase		 
dbHost		 
dbpass		 
dbUsername		 
emailPassword		 
emailUname	team.prometheus@willowmoore.co.za	 
KEY	VALUE	

 Personal ▾ cos301db prometheuspuzzles

1 app ▾

STAGING

 prometheuspuzzles

- Select the Deploy tab and select a branch to deploy. After which select "Enable Automatic Deploys" to enable automatic deployment when you push to the selected branch.

The screenshot shows the Heroku dashboard for the 'prometheuspuzzles' app. The 'Deploy' tab is selected and highlighted with a red circle and a red arrow labeled '1'. Below the 'Connected to a pipeline' section, the 'Deployment method' is shown as 'Heroku Git' (Use Heroku CLI) and 'GitHub Connected' (Use Heroku CLI). The 'Automatic deploys' section is expanded, showing a message about changing the main deploy branch from 'master' to 'main'. Below this, the 'Choose a branch to deploy' dropdown is set to 'Website' and highlighted with a red circle and a red arrow labeled '2'. The 'Wait for CI to pass before deploy' checkbox is checked. At the bottom of the 'Automatic deploys' section, the 'Enable Automatic Deploys' button is highlighted with a red circle and a red arrow labeled '3'. The 'Manual deploy' section is also visible at the bottom, showing a 'Deploy a GitHub branch' option with a 'Deploy Branch' button.

2.1 Setting up your database

You will need to have the following tables in your database:

- testPuzzleRatings

- testPuzzle
- testSolveAttempts
- testUsers

Each table will need the following columns:

1. testPuzzleRatings
 - rating: type: DOUBLE,
 - puzzleID: type: NUMBER,
 - userID: type: NUMBER
2. testPuzzle
 - name: type: STRING,
 - description: type: STRING,
 - puzzleObject: type: STRING,
 - creatorID: type: INTEGER,
 - shared: type: BOOLEAN,
 - image: type: STRING
3. testSolveAttempts
 - solverID: type: INTEGER,
 - puzzleID: type: INTEGER,
 - solved: type: BOOLEAN,
 - attempted: type: BOOLEAN,
 - createdAt: type: DATE,
 - attemptDuration: type: STRING,
 - bestTime: type: STRING
4. testUsers
 - name: type: STRING,
 - username: type: STRING,
 - password: type: STRING,
 - token: type: STRING,
 - verified: type: BOOLEAN

2.2 Customizing your server

2.2.1 Changing server's default port

1. If you wish to alter the default server port on localhost, you may do so by modifying the file: **/bin/www** and locating the variable **PORT**

2.2.2 Adding your web pages to be deployed

1. All additional web pages and subsequent CSS and JavaScript files intended for the use of the user should be placed in `/src /src/app` (see **Section 2** for usage of the front-end and Angular components)
2. In order to access the API the following routes are used:
 - `/routes/user`
 - `/routes/puzzles`
 - `/routes/api`
3. By default all links to the API (in the front end Angular files `/src/app`) will point you to the API hosted on Heroku.

3 Angular Usage - Front-end

The following documentation was generated using Angular CLI:

This project was generated with [Angular CLI](<https://github.com/angular/angular-cli>) version 9.0.6.

3.1 Development server

Run **‘ng serve’** for a dev server. Navigate to **‘http://localhost:4200/’**. The app will automatically reload if you change any of the source files.

3.2 Code scaffolding

Run **‘ng generate component component-name’** to generate a new component. You can also use **‘ng generate directive—pipe—service—class—guard—interface—enum—module’**.

3.3 Build

Run **‘ng build’** to build the project. The build artifacts will be stored in the **‘dist/’** directory. Use the **‘-prod’** flag for a production build.

3.4 Running unit tests

Run **‘ng test’** to execute the unit tests via [Karma](<https://karma-runner.github.io>).

Running end-to-end tests

Run **‘ng e2e’** to execute the end-to-end tests via [Protractor](<http://www.protractortest.org/>).

3.5 Further help

To get more help on the Angular CLI use **‘ng help’** or go check out the [Angular CLI README](<https://github.com/angular/angular-cli/blob/master/README.md>).

3.6 Angular File Structure

We kept the file structure that Angular uses.

3.7 Main Folder

The parent directory consists of all our proxy files, package files, config files and our gitignore file - which consists of our node modules folder and other files git should ignore. It also has the src folder which consists of our whole website.

3.8 src folder

The src folder has the index.html file which is the main file that gets called when our website is running. The src folder also has the assets folder and the app folder.

3.9 assets folder

The assets folder consists of all our images, fonts and our js files used for the manual creation.

3.10 app folder

The app folder has the app component and the routing component. It also has the following folders: models, services, pages, navbar and rate-dialog. The navbar and rate-dialog is used throughout our website and that is why they have separate components on this directory.

3.11 models folder

The models folder is used for the puzzles, puzzleRatings and users to provide a model that we can use throughout the website.

3.12 services folder

The services folder consists of our the service that interacts with our API.

3.13 pages folder

We use the pages folder for all the pages components of our website. Each folder inside the pages folder uses the same file structure - we have the .html file for all the html of the page, then we have the .css file that consists of the styles for that specific page, we also have the .ts file that consists of the functions and we have the spec.ts file that will be used for unit testing.