

Developer Test

Java backend – HTML5, Javascript rich front end

Overview of task

Create a basic single page html5 web application where a user will log in, and using their mouse free hand draw in one of 16 colors, creating drawing strokes on a blank white canvas. Allow the user to move forward and backwards through an Undo/Redo chain, and allow them to save, load, and delete their drawings from hosted server storage.

This should take someone experienced and working with the required full skill set about 10h to complete to a production release quality level, but that is not the goal of this test.

You are free to use any pre-existing solutions for parts of this project. All the answers and training are freely available on the internet for all components of this test. The main goal is to see if you can collect all this knowledge and bundle it up together into a usable product first, and then focus on improving the quality of your code, build pipeline, features, etc.

Where you need to use a quick code hack (ie. working around 3rd party code issues, etc.), make sure to make note of that in code comments with the issue, and how it should be properly resolved with more time.

You will be setup with access to Mindfuel's Gitlab server and a repository that you will have full control over. Check in your code early, and often to show development progress, resolving bugs, etc.

The deadline is the the Monday, 2 weekends after you have been provided this document. A working .war file or tomcat embedded in .jar should be present in the root of the repository for us to test.

Place a readme.md file in the root with any extra details we need to know for making the build, and running the war/jar. ie Environment variable names, etc.

This is not a pass/fail - Complete as much as you can, and as many of the requirements as possible in the time allotted.

The 'testing' operating system will be Debian Jessie (8.x)

Overall requirements

- Your code will be responsible for setting up the database schema.
- You may use any backend storage/database system you want, but we would prefer to see it attaching to a PostgreSQL or MySQL database. If needed, a PostgreSQL/MySQL will be provided with a blank database setup. Host, User, and Password details will be provided to your application via Environment Variables.
- The application will require a login via a web form with a username and password.
- An initial user of Username:admin Password:admin will be pre-seeded.
- The admin user will have functionality to CRUD new users with username & initial password. Only password needs to be updatable.
- Admin UI elements should be available via some icon. When admin elements are not visible, the admin should function as a regular account, with a normal drawing canvas (except for the admin access icon)
- Single page application where possible
- You can have a fixed canvas size ie..800x600 – but the overall UI should scale with browser within reason.
- You can set a minimum browser size, but there should be a graceful fail below that minimum.
- Each 'draw' will be from a mouse press to mouse release
- There should be able to be at least 64 draw actions. How you handle drawing over this limit (if present) is up to you.
- Action of Mouse release, while off canvas action up to you
- A 16 colour pallet should be available in the UI to choose from.
- Line/Point thickness is a 2x2 pixel square. You do not have to provide a way to change this.
- Undo and Redo available for full history to the limit of draw actions above (your choice how to handle over limit)
- Undo only has to remove draw, it does not _have_ to repair damage done by the removal
- Drawings should be able to be saved to the backend database, per user.
- Drawings should be able to be loaded from the backend database, per user.
- Loaded drawings should replay on to a blank canvas with the 'draw' operations in the original order.
- There should be a button to 'Erase and restart' the canvas
- Erasing a canvas with any draw moves should effect a confirmation prompt
- Saved Drawings will require a name. Unique within an individual users set of drawings.
- Loading drawings should present a paginated list of available drawings (for a given user)
- Code will be checked in at various stages into Mindfuel's GIT repository, and you will be setup with an account for that.

Backend Requirements

- Java application based on Spring Boot
- Eventually output to a WAR file that will be run as a ROOT.war package by an external tomcat or
A jar file with an embedded tomcat
- Maven for build manager

- Database is up to you, but we would prefer to see PostgreSQL or MySQL if you have listed those as qualifications.
- Any database used, if not embedded, will need to run under Linux, and be able to be deployed by Mindfuel for testing without requiring any purchase of a license, or registration process.
- Database access details will be provided via OS environment variable(s)
- Appropriate Unit testing
- Backend to Frontend data interchange format will be json

Frontend Requirements

- Angular JS 1.5/2 will be used to provide quick user interaction, and mostly a single page / api driven single page application.
- HTML 5 Canvas Object
- Will work on both Chrome and Firefox of latest public release
- Appropriate Unit testing
- Some Integration/Functional Testing should be attempted
- CSS Bootstrapping of your choice, but we will expect a moderately scalable UI
- Function over form

Resources

<http://www.williammalone.com/articles/create-html5-canvas-javascript-drawing-app/>

<http://intridea.github.io/sketch.js/>

Mindfuel GIT server – <https://git.mindfuel.ca>

Questions / Issues / Clarifications ? Contact richard@kerkhofftech.ca