Coding Challenge

1. About Us

ZyeLabs is a company that is focused on creating exponential value. We partner with organizations and use data, software and design to create software solutions that solve their most pressing business problems.

2. About the Challenge

This coding challenge is all about problem solving and coding - so it should be fun! The purpose of this is to determine how you go about solving problems, how quickly you can learn and determine if you can ship working code.

This challenge is unsupervised, and you can do it in your own time. The coding challenge needs to be completed in **1 day / 24 hours** from the time you receive this.

FAQ

What if I am not done in time?
This is perfectly ok, just send what-ever you have at the point when the time is up. Remember this is a POC

What programming language can I use?
You can use any programming language. Select the one you are most comfortable with.

Should I use a web framework, if so what?
A web framework may speed things up if you are familiar with one. If not familiar with a web framework we would recommend using the one that simplest to get started with. If you are familiar with PHP that could be an option as well. Some Frameworks (You do not have to use these, they are just examples):

- Python: https://palletsprojects.com/p/flask/
- Ruby: https://github.com/sinatra/sinatra
- Javascript: https://expressjs.com/

Where should I deploy the application?

You can deploy it anywhere you like. Some platforms that support quick and easy deployment include:

- https://www.heroku.com/
- https://try.digitalocean.com/

Scoring of the Challenge

The challenge will be scored according to the following dimensions:

- Ability to get something working
- Approach to solving the problem
- Quality of code
- Respecting the submission deadline
- Deploying the code to a production server

Submitting your work

In order to submit your work please send an email to hello@zyelabs.net with the subject line: Coding Challenge – Full Name

- Instructions on how to run the solution locally
- Any libraries needed, external components
- A link to the source code
- A link to the actual site
- Ensure you are committing as you make changes to your code

3. The Coding Challenge

Context

The department of education has just finalized and published the results for the 2016 matric results. They would like to build a site to communicate the results in an easy to use manner to the public. You are tasked with building an initial **proof of concept** for this site.

Objective 1: Understand the data

The email sent to you includes a CSV file.

Review the CSV file and try and understand the data. Questions to ask yourself:

- What information would be important to the public?
- Are there any other calculations that can be done on the data to draw further insights?
- What are the trends?

- Are there multiple levels that the data can be reported on?
- Analyse the data with graphs, trend lines

Key Details about the Data:

- Name is the name of the school
- There are 2 columns for each year of matric results:
    - Wrote_{year} – This is the total number of students that wrote the exam in that year
    - Passed_{year} – This is the total number of students that passed for that year

Outcomes:

- Understand the data
- Analyse the data
- Identify additional calculations possible on the data
- Note if you can report on it at multiple levels

Objective 2: Build a Model of the data to store in a database

Build a data model and/or database table for the data from objective 1. Look at the data and your analysis of the data and design a data model for it. Use your selected web framework to build the data model. Make sure you commit your code at the end of this section.

Outcomes:

- Initial project source code
- Initial project directory structure
- Data model for matric result History

Objective 3: Expose the data via REST API

Your next objective is to expose the data via a **REST API**. The API should support both GET and Post requests. **Get** requests should retrieve the records from the database. **Post** requests should create new records in the database. Make sure you commit once this section has been completed.

Outcomes:

- Working Get API that retrieves records from the DB
- Working POST API that creates a record from the DB
- Tested Locally

Objective 4: Import the data

Write a script in any language that will read the CSV file and then create a record in the database for each of these. Use the API (POST) to create the records in the DB. Save the script in your directory and ensure you have committed the code.

Outcomes:

- Records from the CSV file should be created in your local DB

Objective 5: Reporting on the data

Develop a solution to report on the data. This should be based on your understanding of the data from objective 1. You may use any mechanism to report on the data. Remember:

- There may be multiple levels the data can be reported on.
- There may be different calculations done on the data that will make it more meaningful.
- There also is a need to make the site easy to understand and visual for the general public.

NB: The users of the site is the general public!

The report should be a one page summary of the high-level report. You get additional points if you have any other views of the data.

You are free to use any javascript charting libraries you feel is useful. Just make sure it is compatible with the framework you have selected. See:

https://blog.sicara.com/compare-best-javascript-chart-libraries-2017-89fbe8cb112d

Also ensure you have committed after this step.

Outcomes:

- A Completed application with a single page showing a high-level summary report

Objective 6: Deploy Application

Test & Deploy the application to a cloud platform of your choice. Select a cloud provider to host your application, and deploy it.

Outcomes:

- A Completed application with a single page showing a high-level summary report
- Application deployed and available on the internet
- Source code repository that can be shared (i.e. Shared on GitHub)