Course Blocks

INFO263-22S2 - Web Design and Development

Web Design Project Specifications

OpenReview — Employer Review Portal

Introduction

You are required to develop a website for collecting employee reviews about their work experience at various companies. While the expectations about the functionality of this website are pretty basic — at the very minimum there are forms which are meant to retrieve and send data to a database — there's also a Search Engine Optimisation (SEO) component to this project. The core functionality along with SEO steps in your solution will ensure a reasonable grade for this project. And then, to get the top marks you may implement some optional features, such as user registration and authentication, or including JavaScript charts to visualise the data.

The idea for employer review portals has been thoroughly implemented in a website called Glassdoor.com (https://www.glassdoor.com/about-us/) which contains 114 million of employer reviews on 2,2 million employers. The data used in this project, OpenReview, is a small fragment of data available from Glassdoor.com. For inspiration and examples of design you may want to browse through Glassdoor.com, but you may need to create an account (and, possibly, upload your CV) to get to the good stuff, such as review trends and the like.

OpenReview (MySQL) Database

A small database with real-world data (employers and reviews) is provided to help you get started. There are two tables and one view in the database, but you may need to add more tables if you feel this is necessary to implement specific features required for your project. Here's the description of the tables and views in the **open_review** database:

- Table **employer** — the names and some basic details of 18,259 employers.
- Table **employerReview_S** — the details of 27,181 employer reviews for only five (yes, 5) of the listed employers.
- View **reviewedEmployer_S** — the aggregated (average) review statistics based on the 27,181 reviews.

These tables and data are available via three different sources:

- The **open_review schema** on the MySQL server we've been using in the labs.
- The **open_review_s_dump(.zip)** MySQL dump that can be downloaded from the Project Resources folder and imported in MySQL on your local machine.
- The **open_review_s_sqlite.db** that can be downloaded from the Project Resources folder and opened in DB Browser for SQLite (https://sqlitebrowser.org/).

To get a better idea of the data go ahead and explore the databases in one of those three sources/formats. It is essential that you examine the structure of the tables and views — this will give you a clear idea what data is expected when designing SQL insert queries and the pages for say, recording review data. For example, you may want to run this sort of SQL command to show the structure of the employer table:

```
DESCRIBE open_review.employer;
```

Alternatively, you may want to use a statement of this kind:

```
SHOW CREATE TABLE open_review.employer;
```

Similarly, in SQLite table structure may be obtained like this with a dot command:

```
.schema employer
```

It will be very important for you to examine the structure of the **reviewedEmployer_S** view using the **SHOW CREATE VIEW** statement in MySQL and the equivalent statement **.schema VIEW** in SQLite.

Solution Requirements

The core task is to develop a website for collecting reviews and displaying review information. This task will require a bit of front-end work for designing HTML pages with forms and back-end functionality written in PHP code; and then, to give the website a realistic feel you'll need to add some interactivity with JavaScript and CSS. There is a lot of freedom as to how you may approach this task. At the very minimum your solution must include these basic features/properties:

- The start/home/index page for your solution, providing the used and search engine robots/crawlers with the basic information about your site. This page should also include the navigation options for accessing the rest of the pages in your solution.
- The overview page to display the aggregate ratings/values from the **reviewedEmployer_S** view.
- The functionality (pages, PHP code, SQL queries) for submitting a review for one of the listed employers. Please note that due to the number of listed employers (18,259!) you'll have to think of some smarter option for selecting an employer to be reviewed in any given instance. The fields to be include in the reviews (and in the HTML/PHP form) are to be based on the table containing the reviews.
- The SEO features/steps required for improving the ranking of your page on Google.
- The appearance and interactivity (the look and feel) of the website must use appropriate layout, graphics, and interaction flow. You may use suitable JavaScript/CSS libraries/frameworks to style and enhance your solution.

For the advanced features you may choose to implement some or all of the following:

- Implementing login pages, and the features for creating user profiles/user registration, with appropriate features. This should use appropriate security precautions and avoid storing plain-text passwords in the database.
- Implementing page(s) with graphs to display statistics for individual employers based on the reviews. To see some examples of basic visualisations explore the Glassdoor.com website. Needless to say, it recommended that you use a JavaScript library for including/rendering graphs in your solution; for example, Chart.js (https://www.chartjs.org/) is a good choice for this project, but you may choose some other suitable library, or even write your own. Here's a page how to include Chart.js it in your project (https://dyclassroom.com/chartjs/chartjs-how-to-draw-bar-graph-using-data-from-mysql-table-and-php).
- Implementing page(s) with overall time trends for the listed employers. This task will require some clever SQL queries to generate data (based on reviews for a given employer) to display temporal trends in charts/graphs. Again, you may want to explore how these features are implemented on Glassdoor.com.

You solution must be accompanied by a four-to-five page documentation/report presenting/describing the details of your solution. This documentation is essential to guide those who will be marking your solution — please include the information to help the marker gain a clear understanding of your project solution.

Please note that the database-supported functionality may be implemented either using MySQL (your local MySQL server or the server we've been using in the tutorials) or you may choose to use SQLite database and data, which will be simple files in .db format that you can incorporate into your solution using our good friend PHP Data Objects' implementation of the SQLite driver, PDO_SQLITE. — the SQL queries and the rest of functionality will be pretty much identical, whether you choose to use MySQL or SQLite. The project skeleton provided along with the data uses SQLite .db because it's too easy to connect a PHP script to an SQLite database — no server connection, no username, no password.

Marking Schedule

The core tasks have 85% allocated to those, which is all the way up to the lower boundary of an A+, so to get into the A+ territory you'd want to think of the advanced features.

- Documentation/report — 25%. The documentation should be sufficient for testing and understanding your solution, presented in proper format and language.
- Basic features — 35%. The pages for showing the website overview (1), aggregate employer ratings (2), employer review page/form (3) with appropriate navigation, structure, presentation.
- SEO features — 25%. The work demonstrating your understanding of the SEO techniques applicable to the scope and context of this project.
- Advanced features — 15% (each). You may choose to implement just one of those advanced features or all three — best solutions will be noted and acknowledged. And before you ask, no, you can't get more than 100% for this project.

Questions and Clarifications via Discussion Forum

Needless to say, you are invited to post questions via our Discussion Forum, and please be sure to discuss in the form anything related to this project, short of sharing your solutions.

Last modified: Wednesday, 14 September 2022, 10:43 AM

◄ Midsemester Test

Jump to...