Overview:

The goal of the test project is to find out how suitable the candidate is for completing tasks and solving problems that the clients pay ██████████ to solve. Candidate is tested against industry-standard requirements for knowledge and skills in the field of web technologies and his or her methods in solving problems related to that field. This test project is suitable for junior developers and above.

██████████ uses a lot of open-source software, libraries and frameworks, but it is important for every developer of the team to be well-versed in native PHP and JavaScript programming. This is why the focus of this test project is on the very basics and fundamentals of development.

Test project itself is primitive and straightforward in nature and is similar in difficulty to a university exam. Project helps determine if the candidate fills the minimum that is required to be or become a very successful web developer.

Test project is only one part of applying as a developer in ██████████. Other things that are taken into account are education, previous experience and the interview with one or more core team members. Depending on candidate the test project either happens before or after the interview.

If there are any questions or if you encounter problems during the project, then you can ask for advice or feedback from team members of ██████████

Details of the project are described below. Drink a lot of water and happy coding!

Technical focus points of the project:

- PHP (+XHTML)
- JavaScript (+AJAX +jQuery (optional))
- MySQL
- Filesystem
- Session-handling (+Security)
- Standards (+W3C +PSR-1/PSR-2)
- Documentation (+Comments)

Task:

To build a simple web-based software that consists of two HTML views, two database tables and AJAX-based authorization and logging.

Required components:

- **View 1 - *login.php* -** This is a PHP script that generates an XHTML page of a log-in form. Form should include a username and a password. If the user clicks on the log-in button, then the software should make an AJAX request to a PHP script **ajax.php**. If the authentication is correct, then the script starts sessions and redirects users to movies.php page. If not correct, then shows an error message to the user.
- **View 2 - *movies.php* -** This view should only be accessed when the user is logged in. View consists of an XHTML page and a table that shows all the movies from the MySQL table **movies** that belong to the currently logged in user. Before the table is a search box and a search button that can be used to find a specific movie from the table (this acts like a filter). In the middle of the page is a table with movie names and the release years. After the table is a form that consists of a text field for a movie name and a dropdown for a movie release year and a submit button. If this form is submitted, then a new movie is added to the database for current user. The data should also be validated to make sure

that the movie name is not empty and the year is within a correct range (1900 until today). You can make the movie submit and search forms work either with AJAX or with regular form submits, this is up to you. In the end of the page is a log out button and if this is clicked then the user is logged out and redirected back to **login.php** page.

- **CSS - *style.css*** - The previous two views have to use the same CSS stylesheet. We do not rate the design skills of the developer and the design of the page can be anything you want, but you should define fonts, background colors and button styles in the CSS.
- **AJAX - *ajax.php*** - This is the script that is used for handling AJAX requests, such as for logging the user in and out. It is optional to also use AJAX for adding movies to the database. When a user logs in or logs out, then the script must write a new log file in the filesystem in a separate folder, for example **/log/session.log**, which stores the username that logged in or out and the time when that happened.
- **MySQL table - *users*** - This table stores user ID, username and password fields. Passwords cannot be stored in plain-text format. Creating new users is not required in the test project, but you should create at least one user with username **admin** and password **adminadmin** so that the system can be tested.
- **MySQL table - *movies*** - This table stores movie name and the release year and the ID of the user who added the movie. Also pay attention to how fast your database would work (especially in search) if it would have millions of movie records stored in the database table.

Things to pay attention to

- It is more important **how the code is written and how it works from security standpoint** than the functionality itself.
- The application should work in **all modern web-browsers**. (*Chrome, Firefox, IE7+, Opera, Safari*).
- It is not necessary to strictly follow PHP PSR-1 and PSR-2 standards, but what we expect is that the **code is written consistently and following good general coding standards** that makes it easier to implement you in teamwork.
- The test project is easy enough that **it is not allowed to use CMS software, external libraries and frameworks to complete it**. The only exception is jQuery, which is

allowed due to it being used extensively due to cross-browser support. We are rating primarily your system design and native PHP, HTML, CSS and JavaScript skills.

- HTML must validate as **XHTML 1.0 Strict** standard in a W3C validator.
- Application must entirely support **UTF-8** for all the data, from movies and usernames and functionalities such as search.
- **Variables and function names and comments must be written in English**. A separate ReadMe or documentation is not required, the code must be written well enough to be understood by other experienced developers.
- It is recommended, but not required, to use **object-oriented programming** and create classes for movies and users.
- You should consider having a **separate configuration file** for storing database access and translations used in the system.
- If some things are confusing to you within this test project, **do not worry**. Based on what you do and complete we will also be able to see your potential for improvement.
- **Do not worry about how long it takes to complete this project**. Some developers complete it in a few hours, some take an entire day and we have hired both.