Technical test – Cecropia Solutions – Mid .Net web developer

Required technologies: Asp.Net (C#), HTML, Javascript (jquery, ajax), git.

Delivery : Via Github repository.

Description

The goal of the project that the candidates demonstrate their skills and knowledge on the relevant technologies for the position, and also the ability to research and learn in case its needed. The project must include a README file with any special instructions or requirements that are relevant to be able to evaluate the project.

Part I – RESTful API (50%)

You must implement a RESTful web service using ASP.net that will authenticate an user and will retrieve basic information about it (profile information). Communication will be done using json data, both for sending data to the server and for receiving. It is not necessary that there is a connection to a database. Data can be hardcoded and kept in the server's memory.

There will be 3 public endpoints, specified in the following table:

URI	**Method**	**Input data**	**Output/Results**
api/login	POST	username, password	A json with the result (true/false) and a token (or error message).
api/profile/*token*	GET	Only the token in the URL	A json with basic profile data (name, lastname, email, phone, address).
api/logout/*token*	DELETE	Only the token in the URL	A json with the result (true o false).

The *login* call will send a json message by POST with the username and password (can be plain text) to be validated. If the user doesn't exist, or the password is incorrect, it should return false and an appropriate message. If the validation succeeds, it should return a token, that will be an alphanumeric string randomly generated by the server and kept in the memory. For example:

{“result”: “false”, “message”: “user does not exist”}

{“result” : “true”, “token”: “111000999ABCD”}

The *profile* call will receive a token in the url. The server will verify that the token is valid (previously generated during the login) and will retrieve a json with the profile data listed above. If the token is invalid, it should return an appropriate error.

The *logout* call will receive a token, the server will validate it and will erase it from its memory, returning a json with the result. If the token is invalid, the server will return false and an error message.

In every case, if the endpoint is called with a method different than the one that is accepted, the error should fail with a HTTP 405 error.

Part II– API consumer / web client (40%)

The second part of the project will be a client web application that consumes the API created in part I. For this, you will use **HTML** and **Javascript** alone. Communication should be asynchronism using Ajax (you may use Jquery). The GUI doesn't have to be very elaborated, it can be basic HTML and error notifications may be simple alerts, but a good user experience and attention to details will be appreciated.

With this client application, the user should be able to login. If the login succeeds, the application should call the endpoint to get the profile data, and show it to the user in an appropiate way (which can be a simple HTML table). While being logged in, the user may log out at any moment.

Delivery (valor 10%)

The project should be uploaded into a Github repository. You must have a valid Github account. The name of the repository will be sent to your email address.

You must commit and push your source code to the repository (you may have all your project on master). The last commit should be done before the project due date and time.

Extra Credits

Keep the data in a database (relational or not) and use an MVC framework (both for backend and frontend)

References

http://www.asp.net/web-api

http://docs.kendoui.com/tutorials/ASP.NET/asp-net-hello-services