Zalex Inc. Employee Certification Solution

Zalex Inc. is a multinational organization specialized in manufacturing and selling of Kamado-style grills. The company has locations in over 30 geographies around the world involving either manufacturing facilities or sales offices.

Zalex Inc. uses FAB Human Resources Management platform allowing employees to view their HR files, attendance information, leaves etc.; however, the solution lacks ability to obtain certificates of employment which employees often need to perform different formalities. This is adding overheads to the organization's human resources team and line managers who must manually prepare or approve the letters.

As the result of the above, the organization decided to proceed with an initiative to automate the process by which certificates are obtained. However, to prove benefits early on, the organization would like to proceed with a minimum viable product (MVP) before scaling the solution.

The following is a product backlog for this MVP that you are expected to implement.

Business objective

Simplify the process by which employees obtain their certificate of employment.

Features

#	Feature	Area
F01	**Solution to run as a web-based application** Application is web-based accessible through browser and built using React.	**Architecture**
F02	**Request certificate** Users can submit and validate their certificate requests.	**Business**
F03	**List all relevant certificates** Users can list, sort, filter their certificate requests.	**Business**
F04	**View individual certificate request** Users can expand a request to view all details	
F05	**Update a certificate request** Users can make minor amendments to the requests	

Solution Requirements

F01 - Solution to run as a web-based application

#	Requirement
F01-R01	**Web-based application runs React** The web-based application should use React as the main technology to build the front-end application. No requirement to use advanced React frameworks such as Next.js.

F02 - Request certificate

<table>
<tr><th>#</th><th>Requirement</th></tr>
<tr><td><b>F02-R01</b></td><td><b>User can indicate the requirements of the certificate</b>

<b><i>As an</i></b> <i>employee</i>
<b><i>I want to</i></b> <i>provide information relevant to the certificate I need</i>
<b><i>So that</i></b> <i>the generated certificate has the information needed for the purpose</i>

User can access the “Request Certificate” to fill details for the certificate. The details expected are:

<table>
<tr><th>Field Name</th><th>Type</th><th>Restrictions</th><th>Required</th></tr>
<tr><td>Address to</td><td>Text area</td><td>Alphanumeric</td><td>Yes</td></tr>
<tr><td>Purpose</td><td>Test area – with styling</td><td>Minimum 50 characters</td><td>Yes</td></tr>
<tr><td>Issued on</td><td>Date field</td><td>Future dates only</td><td>Yes</td></tr>
<tr><td>Employee ID</td><td>Text field</td><td>Numeric only</td><td>Yes</td></tr>
</table>
</td></tr>
<tr><td><b>F02-R02</b></td><td><b>User sees error messages if validations fail</b>

<b><i>As an</i></b> <i>employee</i>
<b><i>I want to</i></b> <i>view fields having data issues</i>
<b><i>So that</i></b> <i>I can easily recognize what to change</i>

System should validate the data provided by the user, based on the restrictions in F02-R01, and display error message indicating the field having issue. The validation should happen in-line and prior to submitting the form.</td></tr>
</table>

F02-R03 **User requests are submitted to backend**

Upon submitting, application should call a REST endpoint to create the request on the backend.

Endpoint: `https://zalexinc.azure-api.net/request-certificate?subscription-key=<INSERT API KEY HERE>`
Authentication: API Key
Method: POST

Expected request body (Example):

```
{
    "address_to": "Embassy of Neptun",
    "purpose": "Visa Application",
    "issued_on": "12/9/2022",
    "employee_id": "123456"
}
```

Expected response body (Example):

```
{
    "responce": "Ok"
}
```

F02-R04 **User receives a confirmation message when request is submitted**

As an *employee*
I want to *view a confirmation message once my request is submitted*
So that *I can be sure that the request was successful*

When the user presses the “Submit” button, if the submission is successful (Response is “Ok”), then application should display a message confirming this.

F04 - List all relevant certificates

<table>
<tr><th>#</th><th>Requirement</th></tr>
<tr><td><b>F04-R01</b></td><td><b>Users can view a list of all submitted requests</b>

<b><i>As an</i></b> <i>employee</i>
<b><i>I want to</i></b> <i>view a list of all requests I submitted</i>
<b><i>So that</i></b> <i>I can easily see them all in one place and know the status</i>

User can access the “Requests List” to view the requests. A table view is presented with the following columns:

<table>
<tr><th>Column Name</th></tr>
<tr><td>Reference No.</td></tr>
<tr><td>Address to</td></tr>
<tr><td>Purpose</td></tr>
<tr><td>Issued on</td></tr>
<tr><td>Status</td></tr>
</table>
</td></tr>
<tr><td><b>F04-R02</b></td><td><b>Users can sort the requests list</b>

<b><i>As an</i></b> <i>employee</i>
<b><i>I want to</i></b> <i>be able to sort the requests</i>
<b><i>So that</i></b> <i>I can find my requests in the order I prefer</i>

User can use the headers of the table to sort the data in ascending or descending order.

<table>
<tr><th>Field Name</th><th>Sorting</th></tr>
<tr><td>Reference No.</td><td>Not applicable</td></tr>
<tr><td>Address to</td><td>Not applicable</td></tr>
<tr><td>Purpose</td><td>Not applicable</td></tr>
<tr><td>Issued on</td><td>Yes</td></tr>
<tr><td>Status</td><td>Yes</td></tr>
</table>
</td></tr>
</table>

F04-R03

Users can filter the requests list

As an *employee*
I want to *be able to search the requests*
So that *I can locate specific requests quickly*

User can use an in-line filter to quickly filter the table based on specific combination of multiple fields.

Field Name	**Filter**	**Filter type**
Reference No.	Yes	Full match
Address to	Yes	Contains words
Purpose	Not required	
Issued on	Not required	
Status	Yes	Full match

F04-R04

Requests are retrieved through an API call

Upon accessing the “Requests List” page, the application should call a REST API and obtain all the user requests. This data can then be populated in the table displayed to the user.

Endpoint: `https://zalexinc.azure-api.net/request-list?subscription-key=<INSERT API KEY HERE>`
Authentication: API Key
Method: GET

Expected request body: N/A

Expected response body (example):

```
[
    {
        "address_to": "Embassy of Neptun",
        "purpose": "Visa Application",
        "issued_on": "12/9/2022",
        "employee_id": "123456"
    }
]
```

F05 - View individual certificate request

<table>
<tr><th>#</th><th>Requirement</th></tr>
<tr><td><b>F05-R01</b></td><td><b>Users can open an individual request from the list</b>

<b><i>As an</i></b> <i>employee</i>
<b><i>I want to</i></b> <i>view the full details of a specific request</i>
<b><i>So that</i></b> <i>I can view and focus on that request only</i>

User can click on an icon placed on the last column of the table to open a full-screen dialog displaying all the basic information of the request.

<table>
<tr><th>Field Name</th><th>Condition to display</th></tr>
<tr><td>Reference No.</td><td>Always</td></tr>
<tr><td>Address to</td><td>Always</td></tr>
<tr><td>Purpose</td><td>Always</td></tr>
<tr><td>Issued on</td><td>Do not show if status is not “<b>Done”</b></td></tr>
<tr><td>Status</td><td>Always</td></tr>
</table>
</td></tr>
<tr><td><b>F05-R02</b></td><td><b>User can view the certificate on the screen</b>

<b><i>As an</i></b> <i>employee</i>
<b><i>I want to</i></b> <i>view an embedded version of the certificate</i>
<b><i>So that</i></b> <i>I can preview before downloading it</i>

A basic document viewer should be placed on the right side of the dialog so that the document is displayed there. All certificates are issued in PDF format.

• If the certificate is issued, display the certificate PDF file.
• If the certificate is not issues, display “Certificate is yet to be issued.”

There is no need to annotate or edit the certificate in anyway. The browser PDF viewer is sufficient.

<b><i>IMPORTANT</i></b>
<i>For the sake of the assessment, determine the best way to store and/or display the PDF file. We are not providing an API endpoint that returns the document.</i></td></tr>
</table>

F06 - Update a certificate request

#	Requirement
F06-R01	**Users can update the purpose field** ***As an*** *employee* ***I want to*** *be able to change the purpose of the unprocessed requests* ***So that*** *I can correct the purpose if needed* User can modify the purpose of the request only if the request is in “New” status. Otherwise, field should not be editable. The change should take effect upon confirming them by pressing a button.
F06-R02	**Updated purpose is reflected on the requests list** ***As an*** *employee* ***I want to*** *see the new purpose I put reflected directly on the requests list* ***So that*** *it is consistent with the dialog* Users can change the purpose as indicated in F06-R01, when the change is made, that change should reflect directly on the list sitting behind the dialog without the need to refresh the screen. ***IMPORTANT*** *For the sake of the assessment, we are keeping this functionality simple and not requiring calling the backend APIs to update the records.*

Other Requirements

#	Requirement
T-01	**Deploy and host the application online** Optionally, the solution should be deployed to a cloud hosting platform and accessible online.

Notes

Organization technology standards

- Front-end technologies: React, React Redux
- Microservices technologies: NodeJS or Java
- CI/CD solutions: Azure Pipeline
- Source control: Git
- API style: REST
- React local dev environment hostname/port: localhost:3000 (Required to avoid CORS errors when calling APIs)

Omitted from this case on purpose

- Authentication mechanisms
- Building back-end APIs
- Live APIs – all APIs are mocks with aim to return success by default
- Performing security testing

API Authentication

You will find the API key in a separate email.

*   *   *