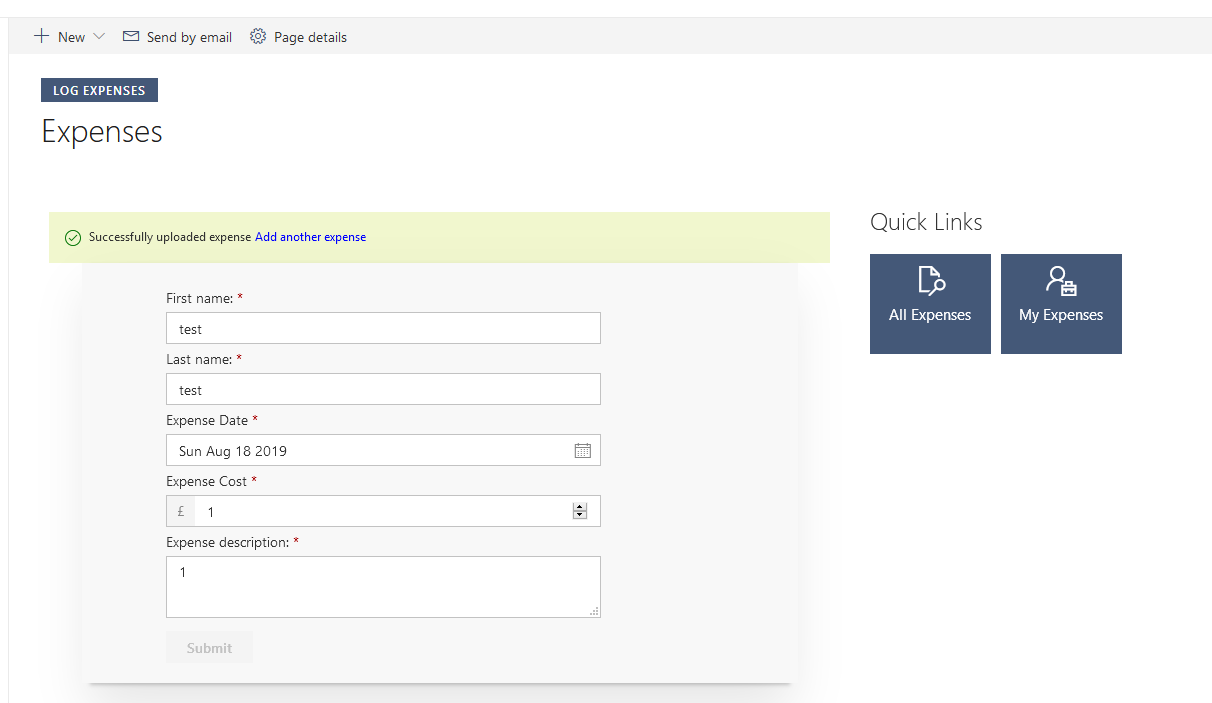
# General information

As I already have an Office 365 tenant, I’ve deployed the SPFx solution.

|  |  |
| --- | --- |
| **Asset** | **Value** |
| GitHub repository | <https://github.com/Fringie/SPFxTechTest> |
| Office 365 username | [test@zainhassan.com](mailto:test@zainhassan.com) |
| Office 365 password | Qaz123!! |
| Site with the web-part & list | <https://zaindev.sharepoint.com/sites/HR/SitePages/Expenses.aspx> |
| CDN Asset location | <https://zaindev.sharepoint.com/sites/CDN/CDN/Forms/AllItems.aspx?id=%2Fsites%2FCDN%2FCDN%2FExpenseClaimsWebPart> |

# Task 1 – Setup Form/list



I’ve implemented this & I’ve kept the form fairly basic. I added some basic form validation and a spinner loading icon which is displayed while the web-part submits the form.

High level overview:

1. Review the requirements
2. Create SPFx solution, add PNP.js
3. Create list & fields in SharePoint manually
4. Create SPFx form & form submit code
5. Deploy solution to the appcatalog
6. Test

# Task 2 – Allow Web-part to work with multiple departments

I added a property to the property pane. The property is the list title; adjusting this property will cause the form to point to another list which satisfies the requirement. Typically I would have added code to generate the list/fields.

# Task 3 – Use Office 365 CDN to deliver assets

1. Create a document library & create a folder within the document library with the web-part name
2. Using PowerShell
   1. Enable CDN
   2. Add CDN endpoint
3. In package-solution.json set includeClientAssets to false
4. Build then package the solution (--ship)
5. Deploy the web-part to the app catalog
6. Copy assets from SPFx solution (temp/deploy folder) to the folder in the document library
7. Add web-part to the page
   1. (Set the web-part property to the name of the list)