

Knowledge Base . . . . .	3
Project Nish Inc. . . . .	4
Documentation . . . . .	8
Business Requirements Document - For Client . . . . .	9
Guidelines . . . . .	12
Requirements . . . . .	16
Action Items . . . . .	18
Design . . . . .	21
Tech Stack Document . . . . .	22
Architectural Diagram . . . . .	24
Proof Of Concept - Frontend . . . . .	25
Proof Of Concept - Backend . . . . .	31
Initial Sketch . . . . .	35
Figma Design Guidelines . . . . .	36
Style Guide . . . . .	38
Sprints . . . . .	41
Sprint 1 . . . . .	42
Sprint Planning 1 . . . . .	43
Sprint Review 1 . . . . .	44
Sprint Retro 1 . . . . .	45
Sprint 2 . . . . .	47
Sprint Planning 2 . . . . .	48
Sprint Review 2 . . . . .	49
Sprint Retro 2 . . . . .	51
Sprint 3 . . . . .	53
Sprint Planning 3 . . . . .	54
Sprint Review 3 . . . . .	55
Sprint Retro 3 . . . . .	56
Meetings Page . . . . .	58
Weekly Standup Meetings . . . . .	59
Weekly Meeting 1 . . . . .	60
Weekly Meeting 2 . . . . .	62
Weekly Meeting 3 . . . . .	63
Weekly Meeting 4 . . . . .	65
Weekly Meeting 5 . . . . .	67
Weekly Meeting 6 . . . . .	69
Weekly Meeting 7 . . . . .	71
Weekly Meeting 8 . . . . .	74
Weekly Meeting 9 . . . . .	76
Weekly Meeting 10 . . . . .	78
Weekly Meeting 11 . . . . .	80
Client Meetings . . . . .	82
Client Meeting 1 . . . . .	83
Client Meeting 2 . . . . .	84
Client Meeting 3 . . . . .	86

Artefacts . . . . .	87
Deployment . . . . .	88
Testing . . . . .	92
Feature Test Scenarios . . . . .	96
API Documentation . . . . .	101
Deployment Guide . . . . .	112
Server NGINX Guide . . . . .	116
Keycloak Realm Configuration Guide . . . . .	119

# Knowledge Base

# Project Nish Inc.

- Executive summary
- Project objectives
- Project scope
- Stakeholders
- Roadmap
- Releases
- Key Links
- Subsections

## Executive summary

'Flemington Kensington News' currently faces several challenges in regard to PDF accessibility and availability on their Squarespace website. Users with disabilities and/or elderly struggle to access the website, in particular viewing PDFs (product). 'Flemington Kensington News' needs to redesign the landing page and the manner in which the Product is stored on the website to allow for accessibility options while also having a smooth, easy-to-use UI.

## Project objectives

By redesigning the website, the aim is to:

1. Create a secondary website with an integrated PDF viewer, with accessibility features such as searchable text, text-to-speech, and document language.
2. Create a searchable feature in Squarespace allowing for PDFs associated with keywords to be located.

## Project scope

In scope for the project:

- Redesign the Squarespace website to include searching.
- Create a secondary website.
- The secondary website must have two faces, Admin View and User View. Admin access allows for uploading, editing, and managing all PDFs. User View will have an integrated PDF viewer that will have options for accessibility features
- Usage of a database to store all PDFs and their associated text files.

This project does not cover:

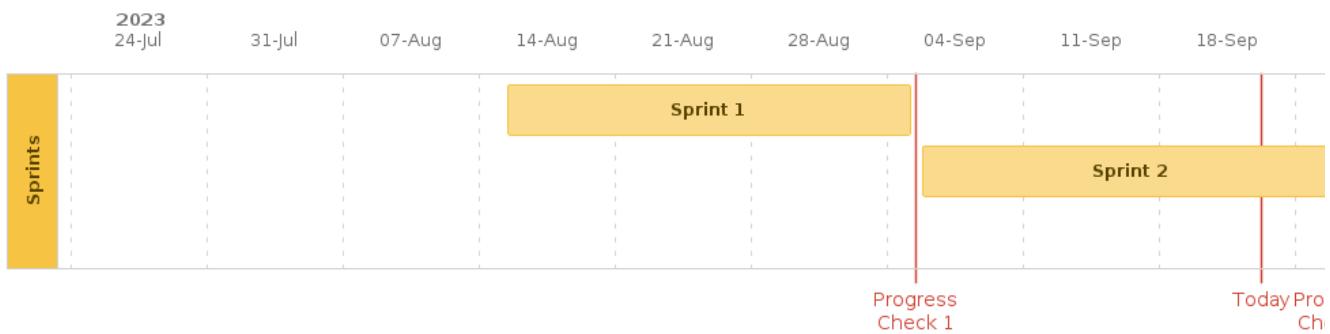
Redesign of Squarespace website in any way other than the above-outlined method

## Stakeholders

Name	Role	Duties
Upnishadd Ksharma	Product Owner	Work closely with the client, and relay information to the team.

Bryn Meachem	Fullstack Developer	Work on the creation of the backend and frontend.
Grace Sendjaja	UI Developer	Work on the creation of the UI.
Matthew Thao Dang Pham	Scrum Master / Backend Developer	Delegation of tasks. Work on the creation of the backend.
Milad Izra	Frontend Developer	Work on the creation of the frontend.
David Fisher	Client	Sets requirements and approves

## Roadmap



## Releases

Release	Frontend Features	Backend Features	Devops Features
<b>17/10/2023:</b> <b>Updated Frontend Endpoint / Overhaul</b>	<ul style="list-style-type: none"> <li>Basic Reader &amp; Further Improvements To Routing [ For Reader Capability]</li> <li>Dynamic Background Colours</li> <li>Rectangles &amp; Ordering On View Page Added</li> <li>Zoom Functionality &amp; Scaling Rectangles with Zoom</li> <li>Updated Endpoints Used instead of previous endpoints.</li> <li>Several Bug Fixes</li> </ul>	N/A	N/A

04/09/2023: Thumbnail Generation on Backend	N/A	<ul style="list-style-type: none"> <li>• Generate Thumbnails on Upload of PDF</li> <li>• Add List all PDF endpoint</li> </ul>	N/A
02/09/2023: (Hotfix) File Sizes	N/A	<ul style="list-style-type: none"> <li>• Allowed Larger File Size through Quarkus</li> </ul>	<ul style="list-style-type: none"> <li>• Allowed Larger File Size through Nginx</li> </ul>
01/09/2023: EC2 Upgrade and Backend Start	N/A	N/A	<ul style="list-style-type: none"> <li>• Upgraded to t2.normal EC2 Instance</li> </ul>
• 9/10/2023: Updated Frontend Endpoint / Overhaul	N/A	<ul style="list-style-type: none"> <li>• Edition Endpoints</li> <li>• Story Endpoints</li> <li>• Selection Endpoints</li> <li>• Minio Resource 2</li> </ul>	N/A
26/08/2023: Initial Frontend Demo	<ul style="list-style-type: none"> <li>• Introduced new Upload Window Modal with drag-and-drop capability</li> <li>• reusable components for modals added</li> <li>• file directory refactored for enhanced organization</li> <li>• sample PDF items now displayable</li> <li>• user feedback animations introduced</li> <li>• New Item Upload Button implemented with click functionality.</li> </ul>	<ul style="list-style-type: none"> <li>• Editions can now be published/drafted. Published Editions cannot be edited. Drafts maintain regular functionality</li> <li>• Searching for Editions added</li> <li>• Keywords Generate for editions</li> <li>• Stories now store html along with delta and text</li> <li>• Selections are now updated as a group of selections instead of individual selections</li> <li>• Delete endpoints for stories and editions</li> </ul>	<ul style="list-style-type: none"> <li>• Started out on t2.micro EC2 instance</li> <li>• Nginx config introduced</li> <li>• Reverse Proxy on port 80 (Forwarded)</li> <li>• Frontend working on Port 3001 through RP (Not Forwarded)</li> </ul>

## Key Links

Software	Link
Discord	<a href="#">Join the IT Project Discord Server!</a>
GitLab	<a href="#">Gitlab</a>
Jira	<a href="#">IT Project   Summary</a>
MinIO	<a href="#">MinIO Console</a>
Backend	<a href="#">OpenAPI UI (Powered by Quarkus 2.16.10.Final)</a>
Prototype Website	<a href="#">Flem Ken News Pdf Viewer</a>

## Subsections

Documentation	<a href="#"> Documentation</a>
Design	<a href="#"> Design</a>
Sprints	<a href="#"> Sprints</a>
Meetings	<a href="#"> Meetings Page</a>
Artefacts	<a href="#"> Artefacts</a>
Deployment Guide	<a href="#"> Deployment Guide</a>
Server NGINX Guide	<a href="#"> Server NGINX Guide</a>

## Documentation

 <b>Business Requirements Document - For Client</b>
 <b>Guidelines</b>
 <b>Requirements</b>
 <b>Action Items</b>

# Business Requirements Document - For Client

---

## PDF TAGGING & ACCESSIBILITY

**2<sup>nd</sup> SEPTEMBER 2023**

---

**Product Owner: Upnishadd Ksharma**

**Document status: Draft**

### Executive summary

'Flemington Kensington News' currently faces several challenges in regard to PDF accessibility and availability on their Squarespace website. Users with disabilities and/or elderly struggle to access the website, in particular viewing PDFs (product). 'Flemington Kensington News' needs to redesign the landing page and the manner in which the Product is stored on the website to allow for accessibility options while also having a smooth, easy-to-use UI.

### Project objectives

By redesigning the website, the aim is to:

1. Create a secondary website with an integrated PDF viewer, with accessibility features such as searchable text, text-to-speech, and document language.
2. Create a searchable feature in Squarespace allowing for PDFs associated with keywords to be located.

### Project scope

In scope for the project:

- Redesign the Squarespace website to include searching.
- Create a secondary website.
- The secondary website must have two faces, Admin View and User View. Admin access allows for uploading, editing, and managing all PDFs. User View will have an integrated PDF viewer that will have options for accessibility features
- Usage of a database to store all PDFs and their associated text files.

This project does not cover:

Redesign of Squarespace website in any way other than the above-outlined method

## Stakeholders

Name	Role	Duties
Upnishadd Ksharma	Product Owner	Work closely with the client, and relay information to the team.
Bryn Meachem	Fullstack Developer	Work on the creation of the backend and frontend.
Grace Sendjaja	UI Developer	Work on the creation of the UI.
Matthew Thao Dang Pham	Scrum Master / Backend Developer	Delegation of tasks. Work on the creation of the backend.
Milad Izra	Frontend Developer	Work on the creation of the frontend.
David Fisher	Client	Sets requirements and approves

## Functional requirements

Requirement Type	Requirement Description
<b>Functional Requirements</b>	
<b>User Authentication and Authorization</b>	The system shall provide secure user authentication and authorization mechanisms with distinct roles: Admin and User.
<b>Secondary Website Creation</b>	The system shall create a secondary website accessible through a separate URL with distinct Admin and User views.
<b>Admin View</b>	Admin users shall be able to log in using unique credentials to access the Admin view for PDF management.
<b>User View</b>	The User view shall be publicly accessible without requiring user authentication and shall offer PDF search capabilities.
<b>PDF Upload and Management</b>	Admin users shall be able to upload PDFs, edit metadata, and replace existing PDFs through the Admin view.
<b>Search Functionality</b>	The User view shall provide a keyword-based search feature for users to find relevant PDFs based on entered keywords.
<b>PDF Viewer with Accessibility Features</b>	The User view's integrated PDF viewer shall offer accessibility features including searchable text and

	text-to-speech.
<b>PDF Database</b>	The system shall utilize a database to store PDF files along with associated metadata and extracted text for search.
<b>Responsive UI Design</b>	The secondary website shall be designed to offer optimal viewing experiences on various devices and screen sizes.
<b>Language Selection</b>	Users shall be able to select the desired language for the text-to-speech feature within the PDF viewer.
<b>Documentation</b>	Comprehensive documentation shall be provided, including user guides for both Admin and User views.
<b>Non-Functional Requirements</b>	
<b>Error Handling</b>	The system shall provide informative error messages and user-friendly error handling mechanisms for user interactions.
<b>Performance Optimization</b>	The secondary website's performance shall be optimized to ensure fast loading times and smooth user interactions.
<b>Business Requirements</b>	
<b>Keyword-Based PDF Locator</b>	The search functionality shall allow users to locate PDFs based on entered keywords, enhancing user experience.
<b>User Roles and Privileges</b>	Admin users shall have elevated privileges for PDF management, while User users shall have read-only access.
<b>Usability Enhancement</b>	The redesigned website shall improve user experience for individuals with disabilities and the elderly, increasing accessibility.
<b>Searchable PDFs</b>	The system shall extract text from PDFs and enable text search within the PDFs, enhancing search functionality.

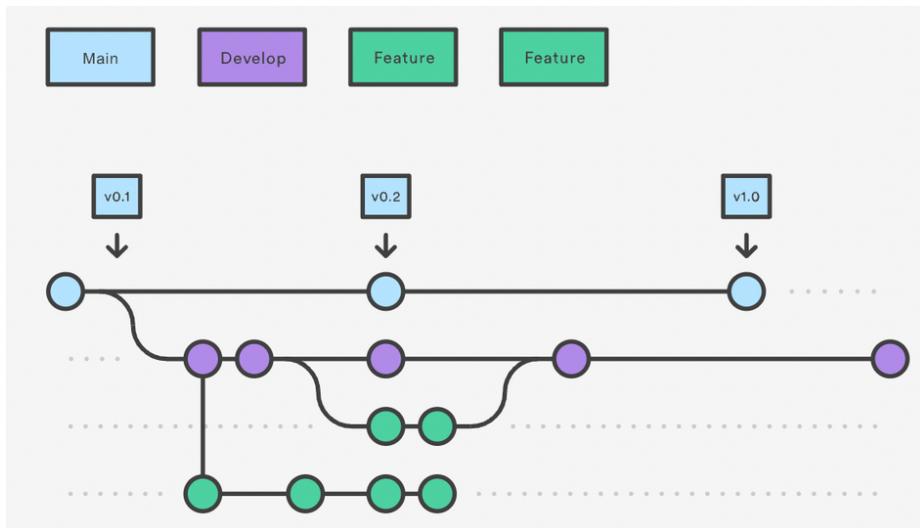
## Guidelines

### Decision Making

The team meets regularly and will make pros and cons for any conflicting perspectives on major decisions. Following a 3 Yes rule, given there are 5 team members, 3 Yes votes will mean a majority decision. Minor stylistic decisions can be determined by respective devs, website decisions can be determined by frontend devs after approval with the team, unless the Product Owner makes an executive decision based on client needs.

### GitLab

#### Git WorkFlow



Each feature will reside in its own independent branch, which after unit-testing and stabilisation will be merged into our developing (Dev) branch. The Dev branch will be our latest stable branch whose features should be forked and rebased when merging back in. Concurrently, our Main branch, will be a live and stable branch that is released periodically in releases. Releases will occur after dedicated and rigorous testing is conducted on our Dev branch to ensure no bad patches are released into the live code. Features will never be merged directly into the main branch.

- All Pull Requests must require at least one approver before merging into Dev branches.
- If merging into Master branches, it must require three approvers.
- Never push to Dev or Master. Always create a feature branch to push your code changes to.
- Each feature will require at least one individual branch, preferably each user story should have its own branch.
- Regularly rebase your code, and rebase before any merge.
- Feature branch naming conventions will be as follows:
  - A name beginning with a feat. e.g. feat-example-branch
  - Less than 30 characters, to ensure clean and simple branch names.
  - Appropriate branch names.
  - Separate words in a branch name with a hyphen (-).
- Hotfixes to dev will follow the above naming conventions other than replacing 'feat-' with 'hotfix-'.
- Any fixes/changes to master will require dev to be updated first and a pull request to update master.

### Code-Base

#### Python

- Import all necessary libraries at the top of a file.
- Use absolute file paths for imports.
- Use 4 spaces per indentation level. (Use tab).
- Limit all lines to a maximum of 79 characters.
- Avoid reassigning values to the same variable.
- Use inline comments sparingly.
- Write docstrings for all public modules, functions, classes, and methods.
- Use snake\_case for naming conventions.
- Use appropriate names for variables.
- Modularise code where possible.

## React.js

- Import all necessary libraries at the top of a file. Remove all unnecessary imports.
- Use 4 spaces per indentation level. (Use tab).
- Maintain consistency over the codebase. Preferable if the use Prettier (VSC Extension) is adopted into the codebase to ensure code consistency over different systems.

### Directory Structure:

Please follow the directory structure closely

- `src` Root: Contains `components`, `pages`, and `store` and `hooks`.

### Components:

- Every standalone component gets its own folder within the `components` directory.
- Folder names are PascalCase and match the main component's name.
- Components related to a specific main component are placed inside the main component's folder.
- Example:

```
1 /components
2   /Modal
3     /UploadModal
4       UploadBox.js
5       UploadContent.js
6       UploadFooter.js
7       UploadHeader.js
8       UploadModal.js
```

### Naming Conventions:

- **Components:** Use PascalCase for component names & their filenames
  - Example: `UserProfile.js`
- **Hooks:** Use kebab-case for hook names, starting with the word `use`
  - Example: `use-navigator.js`
- **Redux Slices:** Use kebab-case for slice names, containing the word `slice` within it.
  - Example: `named-slice.js`
- **Variable, Function Names & Props:** use camelCase
  - Example: `const isHovered = ...;`
  - Example: `<Form onSubmit={doSomethingCool}>`
- **Constants:** use `UPPER_CASE` with underscores.
  - Example: `const API_LINK = "https://api.fake.example.com";`
- **Components**
  - Prefer functional components with React hooks unless specific reason
  - Each component should have a **single** responsibility
    - If a component gets too large consider breaking it down into smaller components

### Styling:

- Using Chakra UI for all styling (Subject to change)
  - Refer to Chakra UI documentation for examples and code layout
    - [Chakra UI Documentation](#)

## JavaScript

- Use ES6+ when possible
  - Includes arrow functions, destructuring, template literals.
  - Always use `const` or `let` if needed. Avoid using `var`.

## Java

### General Convention:

- After brief discussion, the backend is happy to work within the [Google Java Style Guide](#) for a baseline convention. That is inheriting all of the guidelines discussed in the Google Java Style Guide
  - In particular:
    - Javadoc must be written for all public facing methods and classes as in the [Section 7.3](#)
    - Javadoc must be formatted as in [Section 7](#)
    - Naming must abide by [Section 5](#)
    - Column limit 100 Characters as in [Section 4.4](#), with consideration on where to separate lines in [Section 4.5.1](#)
  - However explicitly not including:
    - [Section 4.2](#), where the Java Style Guide recommends 2 spaces for block indentation. We will be using 4 spaces for block indentation.
- `String.format` should almost never be used. In particular never for logging. Format Strings in java are reputedly slow and error-prone (mismatched number of arguments)
- Where possible, if a method can be written as a static method it should be written as a static method, as in a lot of cases there is needless overhead that comes with instantiating an object for the sake of calling one method

### Logging:

- Logging should **always** be done with `org.jboss.logging.Logger` (not be confused with `org.jboss.logmanager.Logger`).
  - Where possible avoid adding Loggers to POJOs (Plain Old Java Objects) and place them in Utils/Service/Endpoint classes.
  - Loggers should be instantiated with either:
    - `static final Logger log = Logger.getLogger(Clazz.class.getCanonicalName());` where `Clazz.class` is the Class being logged. (When not working inside a CDI Bean.)
    - `@Inject Logger log;` (When working inside a CDI Bean)
  - `INFO` logging should be kept to a minimum to avoid log bloat. Anything logging specific values of ephemeral/transient variables should be using `DEBUG`
  - Use the `ANSIColour` enum to make specific logs stand out if needed.

#### API Conventions:

- **Pathing:**
  - All paths should be versioned, namely:
    - `v1` for endpoints that are put into production
      - These Endpoints **should not** change their service agreements (must have same input, output, side effects) unless coordinated with the front end.
    - `v2` for endpoints that are still in development
      - These Endpoints are allowed to change their service agreements if needed, as they are still in development. Coordination with the front end may still be needed but is not as much of a necessity as a production endpoint.
  - Unless explicitly needed, specify all path parameter/query parameter inputs for endpoints as final e.g:
    - `final @PathParam("some-path-param") String pathParam`
    - `final @QueryParam("some_query_param") String queryParam`
    - Path Parameters should always be kebab case (`aaa-bbb-ccc`)
    - Query Parameters should always be snake case (`aaa_bbb_ccc`)
- **Headers:**
  - `Authorization` header should be used as in the OAuth 2.0 Standard with Bearer tokens (described in [RFC-6750](#)). Generally speaking:
    - Bearer tokens captured as JWT tokens from keycloak should be put in the `Authorization` header value with the given format: `Bearer <Token>`
    - Should be used whenever sensitive data is being requested/an unsafe http method is being performed
    - The Bearer Token **should** be accessible through a Class-level attribute: `@Inject JsonWebToken`. At the very least, it will be accessible through the `Authorization` Header.
  - Where possible, both Content-Type and Accept headers should be declared:
    - Specify `Content-Type` header with `@Produces(MediaType.XXX)` annotation (`javax.ws.rs.Produces`) (`javax.ws.rs.core.MediaType`)
    - Specify `Accept` header with `@Consumes(MediaType.XXX)` annotation (`javax.ws.rs.Consumes`)
    - Specify all Endpoint Classes as `@ApplicationScoped` (`javax.enterprise.context.ApplicationScoped`)
- **Endpoints that interact with the DB:**
  - Interacting with the db must always happen inside of a Transaction, the easiest way to do this in an endpoint is with the `@Transactional` annotation (`javax.transaction.Transactional`). This creates a Transaction for the scope of the request, and commits it on returning a response. Explicit Transactions should be present for both Reading and Writing to the database.
- **Consuming a Request Body:**
  - Requests bodies are automatically unmarshalled from raw json to the type of the first unmatched parameter in the endpoint's method declaration. This means that if the system is unable to convert the raw json to whatever type it has picked up (Say an Integer) then it will throw an Exception. To reduce the amount of exceptions thrown during this process, the parameter in the method meant to take the body value should be a `javax.json.JsonObject` (not `io.vertx.core JsonObject`, although this would still work).
  - When working with `javax.json.JsonObject`, a `containsKey` check will stop a `NullPointerException` occurring when using calls to `javax.json.JsonObject#get(String key)` or variants of. This family of methods of `JsonObject` will throw an NPE if there is no mapping for the key on the `JsonObject`, so a `containsKey` check saves a bit of overhead and allows optional keys
- **Sending a Response Body:**
  - Response Bodies should be deterministic in their structure. That is, for a given Status Code, the response body accompanying that Status Code should always have the same structure.
    - To support this, whenever adding a mapping to a `JsonObject` to be sent out as a response, if the value would be `null` - instead of omitting the mapping (altering the structure), `javax.json.JsonValue.NULL` should be used in its place to preserve the structure and signify a null value. Simply passing `null` will throw an exception

#### Status Codes:

- As in [HTTP response status codes - HTTP | MDN](#), the general convention for when to send a particular status code should be followed. In particular:
  - Send a 201 Created as a response to POST with a reference back to the unique identifier or the path to the newly created resource as a result of the post in the response body
  - Send a 204 No Content when there is no response body and the result is acceptable.
  - Send a 404 Not Found when the resource specified by the path param cannot be found (not to be confused with when items cannot be found due to no match with query params)
  - Send a 200 OK as a response to PUT with the updated state of the resource being updated if successful
  - Send a 200 OK as a response to GET when the Path Params matches a particular resource but the Query Params put in mean there are no results (typically in filtered searching)
  - Send a 401 Unauthorized when the request fails to produce a valid Bearer Token in the `Authorization` header
  - Send a 403 Forbidden when the request produces a valid Bearer Token as described above, but the Principal specified within the bearer is not privileged enough to execute the requested endpoint
  - Send a 400 Bad Request when the request fails as a result of the inputs being poorly-defined
  - Send a 500 Internal Server Error when the request fails as a result of the processing within the endpoint, but the inputs are well-defined
- **HTTP Methods:**
  - GET should be used when a resource is being requested from the server. A request body **should never** be present in a GET request.
  - POST should be used when the request is asking for a resource to be created. This should generally have a request body, and should return a link back to the newly created resource.
  - PUT should be used when the request is updating a resource. This should have a request body that is all of the existing data of the resource with the parts that are modified. It may either return a link back to the updated resource, or return the new state of the updated resource.
  - DELETE should be used when deleting an existing resource. If the resource is sensitive user data, the data should not be physically deleted from the database (privacy laws) unless specified by the relevant user, but instead marked as "deleted".

- **General Endpoint Logging:**

- A Post-Matching Request Interceptor should be used to do general endpoint request logging. That is logging any of:
  - path
  - request headers
  - request body
  - http method
- If needed, a Response Interceptor can be put in place to do general endpoint response logging. That is logging any of:
  - request headers
  - response headers
  - response body



give cat a kiss before starting new feature, for prosperity.

# Requirements

- Functional Requirements
- Non-Functional Requirements
- Business Requirements

## Functional Requirements

User View	The User view shall be publicly accessible without requiring user authentication and shall offer PDF search capabilities.	Done	High
PDF Viewer with Accessibility Features	The User view's integrated PDF viewer shall offer accessibility features including searchable text.	Done	High
Secondary Website Creation	The system shall create a secondary website accessible through a separate URL with distinct Admin and User views.	Done	High
Search Functionality	The User view shall provide a keyword-based search feature for users to find relevant PDFs based on entered keywords.	Done	Low
User Authentication and Authorization	The system shall provide secure user authentication and authorization mechanisms with distinct roles: Admin and User.	Done	Medium
PDF Database	The system shall utilize a database to store PDF files along with associated metadata and extracted text for search.	Done	Medium
Documentation	Comprehensive documentation shall be provided, including user guides for both Admin and User views.	Done	Medium
Admin View	Admin users shall be able to log in using unique credentials to access the Admin view for PDF management.	Done	High
PDF Upload and Management	Admin users shall be able to upload PDFs, edit metadata, and replace existing PDFs through the Admin view.	Done	High
Responsive UI Design	The secondary website shall be designed to offer optimal	Done	Medium

	viewing experiences on various devices and screen sizes.		
--	--	--	--

## Non-Functional Requirements

<b>Error Handling</b>	The system shall provide informative error messages and user-friendly error handling mechanisms for user interactions.	Done	High
<b>Performance Optimization</b>	The secondary website's performance shall be optimized to ensure fast loading times and smooth user interactions.	Done	Low

## Business Requirements

<b>Searchable PDFs</b>	The system shall extract text from PDFs and enable text search within the PDFs, enhancing search functionality.	Done	High
<b>Usability Enhancement</b>	The redesigned website shall improve user experience for individuals with disabilities and the elderly, increasing accessibility.	Done	High
<b>Keyword-Based PDF Locator</b>	The search functionality shall allow users to locate PDFs based on entered keywords, enhancing user experience.	Done	Medium
<b>User Roles and Privileges</b>	Admin users shall have elevated privileges for PDF management, while User users shall have read-only access.	Done	Medium

# Action Items

## ⚡ Action plan

Action	Owner	Due date	Action items	Ticket
Make sure team is happy and onboard with decisions.	@Upnishadd	26 Sept 2023	Regular informal 1 on 1 contact with team members to ask about progress and thoughts of the project	<p>FIRST-88: Catch up with team member <span style="border: 1px solid green; padding: 2px;">S DONE</span></p>
Define PDF utilities more concretely through a more elaborate PDFWrapper class implementation.	@matthew.thao.pham	26 Sept 2023	Look into Java-Python communication Python Docker Unit tests for PDFWrapper Clean up Python/Java code according to guidelines	<p>FIRST-67: Generate PDF thumbnails <span style="border: 1px solid green; padding: 2px;">DONE</span></p> <p>FIRST-47: Convert the selected section of PDF to text using data extraction techniques. <span style="border: 1px solid green; padding: 2px;">DONE</span></p>
More Integration Testing. Guideline Enforcement	@Bryn Meachem	26 Sept 2023	<ul style="list-style-type: none"> <li>Look into Mockito</li> <li>Write Quarkus Endpoint Tests</li> </ul> Guideline related: <ul style="list-style-type: none"> <li>Focus on Code Smell for a couple days</li> <li>Discuss with @matthew.thao.pham about enforcing guidelines</li> </ul>	<p>FIRST-108: Enforce Guidelines and do some research for backend tools <span style="border: 1px solid green; padding: 2px;">DONE</span></p>
Work on smaller feature branches & More testing for existing features	@Milad Izra	26 Sept 2023	<ul style="list-style-type: none"> <li>Start the new feature branch next week feat-pdf-zoom to implement zooming capabilities [Smaller Feature Branches]</li> <li>Create new Unit Tests &amp; Integration Tests next week.</li> </ul>	<p>FIRST-32: Admins can view the document in either 'PDF' or 'Text' format <span style="border: 1px solid green; padding: 2px;">DONE</span></p>
Reach out to members more often to get feedback and be on the same page. Be more involved in the code	@grace.sendjaja	26 Sept 2023	Add shadows and other details to figma then communicate with teammates	<p>FIRST-24: Discuss and figure out design and design flaws <span style="border: 1px solid green; padding: 2px;">DONE</span></p>

			Make the custom text editor with quill	
Help organise, wrapping up the project	@Upnishadd	16 Oct 2023	Contact team members, arrange and organise what needs to be done from each side of things, and bring together the different parts.	No Ticket Required
Communicate and collaborate with frontend more proactively.	@matthew.thao.pham	16 Oct 2023	Define more concretely what is required of the backend from the frontend and finish those tasks in a timely fashion.	No Ticket Required
Check in with Frontend more often	@Bryn Meachem	16 Oct 2023	<ul style="list-style-type: none"> <li>Ensure that issues in the backend that are picked up by the frontend are not roadblocking the frontend</li> </ul>	No Ticket Required
Set aside time to polish up the backend	@Bryn Meachem @matthew.thao.pham	16 Oct 2023	<ul style="list-style-type: none"> <li>Integrate Keycloak</li> <li>Do SSL</li> <li>Finish Endpoints</li> </ul>	<span style="border: 1px solid #ccc; padding: 2px;">SEC-13: Keycloak Integration</span> <span style="border: 1px solid #ccc; padding: 2px;">DONE</span>
Help out in Frontend	@grace.sendjaja	16 Oct 2023	Contact Milad, discuss which parts need to be completed and start testing and defect fixing.	<span style="border: 1px solid #ccc; padding: 2px;">SEC-2: Frontend Remaining Tasks (In Scope)</span> <span style="border: 1px solid #ccc; padding: 2px;">DONE</span>
Finish Up Frontend Tasks	@Milad Izra	16 Oct 2023	Contact Nish to finalise all remaining tasks and finish off the frontend.	<a href="https://comp30022-nish.atlassian.net/browse/SEC-2">https://comp30022-nish.atlassian.net/browse/SEC-2</a>
Finalise Documentation	@Upnishadd	12 Nov 2023	Clear up any documentation needed to be done.	<a href="https://comp30022-nish.atlassian.net/browse/SEC-47">https://comp30022-nish.atlassian.net/browse/SEC-47</a>
Clean Up Frontend Codebase	@Milad Izra	12 Nov 2023	Clean up codebase, and refactor existing code	<a href="https://comp30022-nish.atlassian.net/browse/SEC-49">https://comp30022-nish.atlassian.net/browse/SEC-49</a>
Document Backend	@Bryn Meachem	12 Nov 2023	Document Backend, ensure Google Style Guide compliant. Write a couple integration tests	<a href="https://comp30022-nish.atlassian.net/browse/SEC-46">https://comp30022-nish.atlassian.net/browse/SEC-46</a>

Support Frontend during API integration phase	@matthew.thao.pham	12 Nov 2023	Be on standby and fix potential bugs that arise during testing	<a href="https://comp30022-nish.atlassian.net/browse/SEC-50">https://comp30022-nish.atlassian.net/browse/SEC-50</a>
Connect Endpoints	@grace.sendjaja	12 Nov 2023	Communicate with backend to finalise the quill page	<a href="https://comp30022-nish.atlassian.net/browse/SEC-48">https://comp30022-nish.atlassian.net/browse/SEC-48</a>

## Design

Tech Stack Document
Initial Sketch
Figma Design Guidelines
Style Guide

# Tech Stack Document

- [React Front End](#)
- [Postgres Database](#)
- [Quarkus Backend](#)
- [Docker/Docker-Compose Deployment DevOps](#)
- [Backend Services](#)
  - [MinIO](#)
  - [Keycloak Authentication](#)
- [Related Content](#)
  - [Proof of Concepts](#)

## React Front End

- **Component-based approach:** promotes reusable code which can aid in achieving a more maintainable and scaleable tool
- **Performance Considerations:** We must consider that users will be accessing the webpage through outdated hardware. Hence, the virtual DOM ensures efficient and smooth user experience.
- **Flexibility:** Easy integration with various backends, lots of existing libraries and APIs that can provide a smooth UI / user experience
- **Tool Usability / Visuals:** Many react libraries often come with built-in animations and interactions, ensuring a modern look and feel.

## Postgres Database

- **Developer Experience:** The team has experience in using Postgres and thus can quickly find solutions to any problems regarding Postgres.
- **Free:** Postgres is open-source, which means that there are no licensing costs associated with its use.  
As the client wishes to reduce cost as much as possible, Postgres is a viable option, with active community support.
- **Object Oriented:** Postgres also allows / supports complex data types. Postgres supports complex types like arrays and JSON, allowing the team to model data in versatile ways.
- **Works with Python:** Postgres offers native support for Python, enabling developers to write database procedures directly in Python.

## Quarkus Backend

- **Developer Experience:** The team has experience in using Postgres and thus can quickly find solutions to any problems regarding Postgres.
- **Efficiency and Speed:** Quarkus is optimised for fast boot times and low memory consumption. When it comes to handling file storage, such as PDFs, this efficiency can significantly enhance user experience, especially when uploading or accessing large PDF files (We won't be, but this efficiency can be helpful, especially for the backend).
- **Unique Identifier for Each PDF:** Assigning a unique ID for each PDF ensures that each file can be quickly retrieved, updated, or deleted without confusion
- **Free / Open-Source:** Quarkus is licensed under the Apache License 2.0. This license allows for the free use, modification, and distribution of the software. Helps client reduce costs of the application. In large scale frameworks, open-source code also allows maintainers of the code to more quickly identify and fix bugs in the code - as they have the assistance of the community using the tech. This goes further to create a vast ecosystem of extensions that can be used within Quarkus.

# Docker/Docker-Compose Deployment DevOps

- **Consistency / Portability:** Docker ensures that the application runs the same, regardless of where the container is being run, reducing "it works on my machine" issues. With docker, deployments become a matter of uploading a precompiled version to the server so it can run the latest, bypassing a lot of platform-specific issues during code compilation.

## Backend Services

### MinIO

- **General Backend Functionality:** Provides Amazon Web Services compatible "bucket" storage. Typically used in Content Delivery Networks (CDN) as it is proficient in both storing and securing large data (things that would not find themselves in a database).
- **Licensed under the GNU AGPL v3 license:** This license allows for the free use, modification, and distribution of the software. Helps client reduce costs of the application. Also provides transparency behind what is going on in the software (minIO is open source).

### Keycloak Authentication

- **Secure Login System:** In a lot of businesses, designing your own password / security framework is risky at best, and is always time consuming. By using an off-the-shelf open source solution, time and resources are saved both in development and security testing, as keycloak is a widely adopted piece of technology that has been tried and tested already.
- **Free & Open Source:** Helps client reduce costs of the application. Open Source also provides transparency about what is going on in the code, which is a key aspect with software that handles user data/credentials, to ensure that the software is not maliciously harvesting them. The code being open source in this instance does not impact the security, as keycloak relies on key cryptographic principles and is designed to handle a number of authentication specifications (OAuth 2.0, OpenID, SAML) without much extra work for the developer.

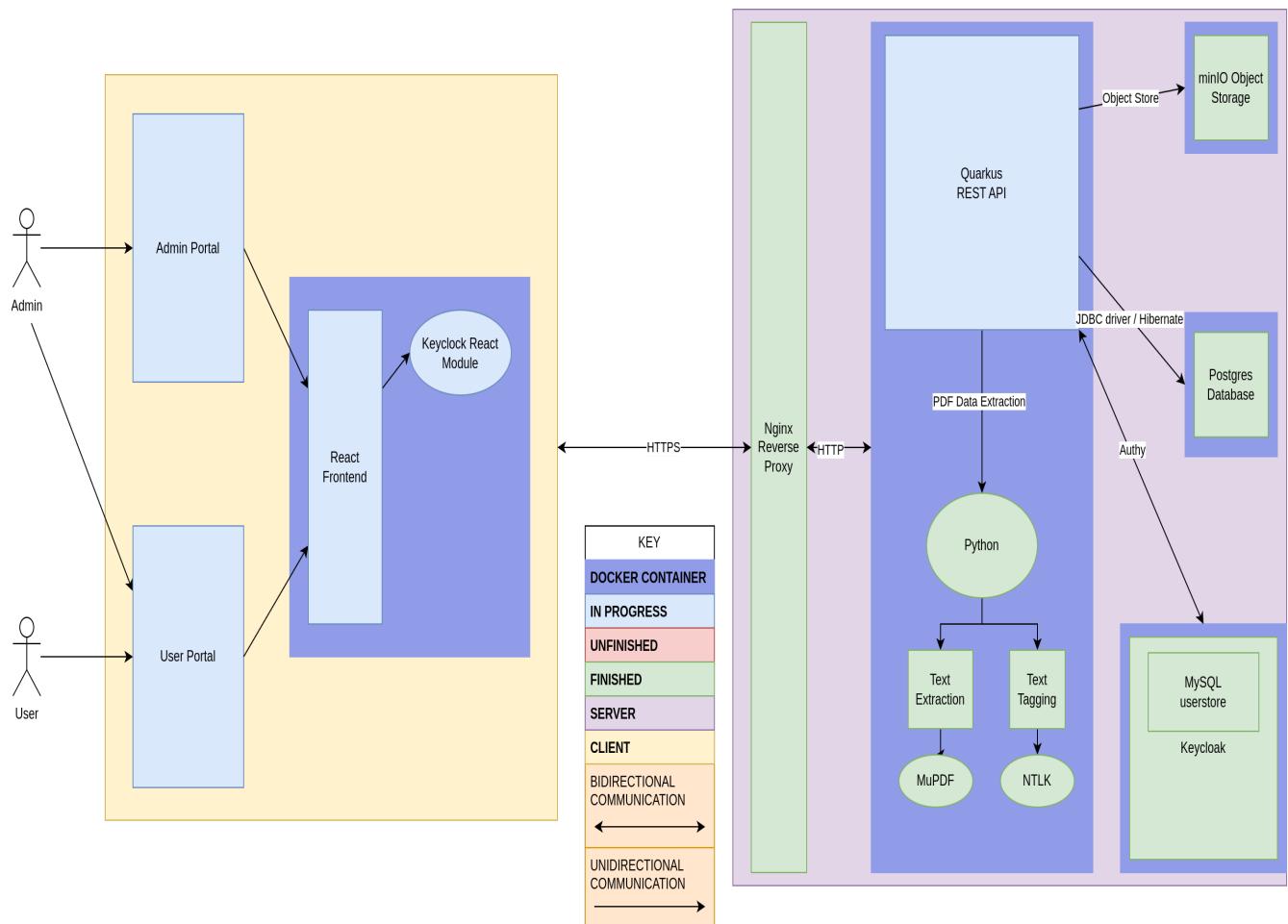
## Related Content

 [Architectural Diagram](#)

### Proof of Concepts



# Architectural Diagram



# Proof Of Concept - Frontend

- Problem Statement
- Required Resources & Technical Solutions
  - React.js 
  - Chakra UI 
  - React Router 
  - React Router 
  - PDF.js & PDF-LIB.js 
- Success Criteria & Technical Indicators
- Scope and Timeline 

## Problem Statement

A long-standing local news agency has a rich 40-year archive filled with scans and PDFs. While this content holds significant value for readers and the community, it's not yet optimized for easy digital access.

In its current form:

-  **Static Content:** Majority of the the content remains stuck in static, non-searchable scans that means readers can not find specific details of articles, stories or events within these PDFs
-  **Lack of Accessibility:** In a digital age, readers expect certain levels of accessibility options to aide user experience, especially those most vulnerable such as zoom, voice reading. The absence of these tools can deter many users from reading, more specifically those who suffer from visual or cognitive disability from accessing the content.
-  **Manual Work:** The clients burden of manging, tagging and categorising such static content without a streamlined system, especially when the content spans over decades can be a insurmountable task to tackle without a helpful tool.

How this tool will help:

-  **Digitization and Search Optimization:** By converting static content and older PDF content into a digitally optimised format, readers will be able to search and locate specific content much quicker than digging through an old archive, and thus increases accessibility.
-  **Integrated Accessibility Tools:** This solution aims to also incorporate seamless essential accessibility tools such as zoom, adjustment of fonts, or utilise text to speech functionalities to cater to all and ensure a universally inclusive reading experience.
-  **Streamlined Admin Tool:** This concept will also aim to offer a simplified way to automate tasks such as tagging and categorising past and future content and reduce manual efforts.

---

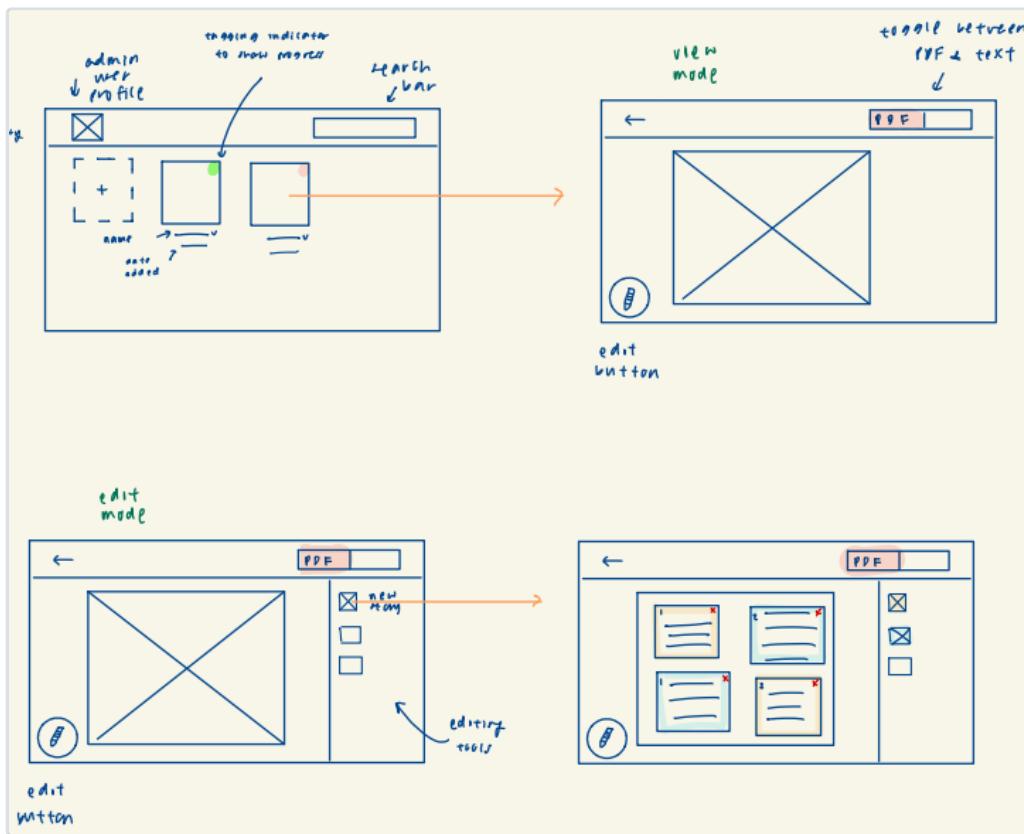
## Required Resources & Technical Solutions

We are adopting a combination of various, popular web technologies and proven libraries. Each component of our tech stack has been selected to address a specific aspect of the problem, ensuring that the final solution is comprehensive and robust.

### React.js

-  **Purpose:** Create dynamic, single-page applications ensuring faster navigation and improved user experience.
-  **Usage:** React will serve as the backbone to the front end, and will power all interface and UI related aspects to users and the client. Its component based approach will help achieve building a single page application for the tool that is easily accessible, that avoids full page reloads.

- 💡 **Transforming Design Concepts with React.js:** The provided wireframe showcases a layout optimized for user experience that is achievable through react. This means seamless transitions, intuitive user flow, and a cohesive design that matches the visual representation laid out in the wireframe.



A preliminary visualization of our tool's layout and user interface, created to ensure an optimal user experience

## Chakra UI 🎨

- 🎯 **Purpose:** Enhance UI development process, ensuring consistent theme and responsive design
- 🔧 **Usage:** Chakra UI will be a necessity in rapidly prototyping, creating and deploying a user friendly user interface for the application. Its diverse component library will provide seamless and intuitive experience for both the user and admin sides. Furthermore, we can leverage the fact that Chakra UI supports code styling through Figma, ensuring a faster development cycles while still maintaining high-quality styling.
- 📋 **Code Implementation & Functionality:** This code snippet below showcases the power of Chakra UI when used with React. This example showcases the simplicity in integrating Chakra UI with react and highlights how the development lifecycle can be simplified, whilst addressing issues of user experience

```

1 import * as React from "react";
2 import { Box, Center, Image, Flex, Badge, Text } from "@chakra-ui/react";
3 import { MdStar } from "react-icons/md";
4
5 export default function Example() {
6   return (
7     <Center h="100vh">
8       <Box p="5" maxW="320px" border="1px solid #ccc">
9         <Image borderRadius="md" src="https://bit.ly/2k1H1t6" />
10        <Flex align="baseline" mt={2}>
11          <Badge colorScheme="pink">Plus</Badge>
12          <Text
13            ml={2}

```

```

14         textTransform="uppercase"
15         fontSize="sm"
16         fontWeight="bold"
17         color="pink.800"
18     >
19         Verified &bull; Cape Town
20     </Text>
21 </Flex>
22 <Text mt={2} fontSize="xl" fontWeight="semibold" lineHeight="sl"
23     Modern, Chic Penthouse with Mountain, City & Sea Views
24 </Text>
25 <Text mt={2}>$119/night</Text>
26 <Flex mt={2} align="center">
27     <Box as={MdStar} color="orange.400" />
28     <Text ml={1} fontSize="sm">
29         <b>4.84</b> (190)
30     </Text>
31 </Flex>
32 </Box>
33 </Center>
34 );
35 }

```



**PLUS VERIFIED • CAPE TOWN**

**Modern, Chic Penthouse with Mountain, City & Sea Views**

\$119/night

★ 4.84 (190)

## React Router 🚀

- 🎯 **Purpose:** Ensure seamless, dynamic navigation within the tool.
- 🔧 **Usage:** React router will provide the application with dynamic routing to maintain that “single-page” app feel. We will primarily utilise React-Routers’ ‘Browser Router’ and ‘Route’; based components to define multiple routes on the single page. Furthermore, React Router provides functionality and necessary hooks to implement redirections after successful authentication. Hence, the router package is a necessity to create a fluid user experience.
- 📜 **Code Implementation & Functionality:** The following code snippet demonstrates the efficiency of React Router when integrated with a React application. This snippet showcases how a single page application can implement seamless navigation.

```

1 const router = createBrowserRouter([
2     {
3         path: "/",
4         element: <PDFUploadPage />,
5         index: true // Set this as the default route
6     },
7     {
8         path: "pdf/:pdfID/tool",
9         element: <PDFRootLayout />,
10        children: [
11            {
12                index: true,
13                element: <PDFToolPage />
14            },
15            {
16                path: "view",
17                element: <PDFViewPage />
18            },
19            {
20                path: "edit",
21                element: <PDFEditPage />
22            }
23        ]
24    }
25 ])

```

```

21           element: <PDFEditPage />
22       }
23   ]
24 }
25 ]);
26
27 function App() {
28     return (
29         <ChakraProvider>
30             <RouterProvider router={router} />
31         </ChakraProvider>
32     );
33 }

```

## React Router

- ⌚ **Purpose:** Efficient global state management.
- 🔧 **Usage:** As we wish to maintain the fluid nature of a single page application, redux can help simplify global states between the tool and the viewer.
- 📜 **Code Implementation & Functionality:** Using redux, we can capture the state of the page and check whether we are in a “view-state” or a “edit-state”. This allows us to more easily ensure the flow of data is manageable and clear. Furthermore, user settings may also be captured through redux to allow for a easier manageable codebase. Using different slices we can also capture different states and keep the codebase manageable by having different files for each store/slice.

```

1 const counterSlice = createSlice({
2     name: "counter",
3     initialState: { counter: 0, showCounter: true },
4     reducers: {
5         increment(state) {
6             state.counter++;
7         },
8         decrement(state) {
9             state.counter--;
10        },
11         increase(state, action) {
12             state.counter = state.counter + action.payload.amount;
13        },
14         toggleCounter(state) {
15             state.showCounter = !state.showCounter;
16        },
17    },
18 });
19
20 const store = configureStore({
21     reducer: {
22         counter: counterSlice.reducer,
23         auth: authSlice.reducer,
24     },
25 });

```

## PDF.js & PDF-LIB.js

-  **Purpose:** Facilitate rendering, tagging and manipulation of PDF-related content within the application.
-  **Usage:** PDF.js will be necessary if we wish to render any PDF content on the application. PDF.js provides many useful tools that can address some problems. It ensures PDF's can be displayed regardless of the users device browser which can help address accessibility concerns. PDF-LIB.js will be also be required as we need to have the ability to manipulate PDF content. PDF-LIB.js has a large arsenal of tools that can be utilised to achieve our requirements all within a browser environment. We can extract the necessary text using these libraries as well and provide users the ability to simply read the text without the need of a PDF viewer.
-  **Code Implementation & Functionality:** Using these libraries, we can achieve all rendering and manipulation purposes related to PDF's.

```
1 import { PDFDocument, StandardFonts, rgb } from 'pdf-lib'
2
3 async function createPdf() {
4   const pdfDoc = await PDFDocument.create()
5   const timesRomanFont = await pdfDoc.embedFont(StandardFonts.TimesRoman)
6
7   const page = pdfDoc.addPage()
8   const { width, height } = page.getSize()
9   const fontSize = 30
10  page.drawText('Creating PDFs in JavaScript is awesome!', {
11    x: 50,
12    y: height - 4 * fontSize,
13    size: fontSize,
14    font: timesRomanFont,
15    color: rgb(0, 0.53, 0.71),
16  })
17
18  const pdfBytes = await pdfDoc.save()
19 }
```

## Success Criteria & Technical Indicators

-  **Efficient PDF Navigation:** Users should have no trouble navigating throughout the application and all it's features. The reader should be fluid and take into account performance and the tool should be easy to use.
-  **Accessibility Standards:** The tool should incorporate accessibility tools that address the needs of those that require such features. They should cater to all and ensure that all readers have the required features and tools to ensure a inclusive reading experience.
-  **Learning Curve:** The admin tool should be simple enough to use that someone new can easily pick up how to use it with minimal training.

## Scope and Timeline

The scope of this project covers the development of two essential components to address the requirements and address issues; the pdf tool and the pdf reader tool. It's necessary that the tool be setup and functional first before the pdf reader tool can begin. It is predicted that the tool will require 1-2 weeks of initial development for the tool and another 2-3 weeks for the accessibility features and reader. This will be followed by a week of testing and feedback to ensure the application is up to standard. The final weeks will address deployment of the project.



# Proof Of Concept - Backend

- Problem Statement
- Required Resources & Technical Solutions
- Quarkus 
  - Tech to be used with Quarkus:
  - Keycloak 
  - Docker 
  - Nginx 
  - minIO 
  - PostgreSQL 
  - Python 
- Success Criteria & Technical Indicators
- Scope and Timeline 

## Problem Statement

A long-standing local news agency has a rich 40-year archive filled with scans and PDFs. While this content holds significant value for readers and the community, it's not yet optimized for easy digital access.

In its current form:

-  **Static Content:** Majority of the the content remains stuck in static, non-searchable scans that means readers can not find specific details of articles, stories or events within these PDFs
-  **Lack of Accessibility:** In a digital age, readers expect certain levels of accessibility options to aide user experience, especially those most vulnerable such as zoom, voice reading. The absence of these tools can deter many users, more specifically those who suffer from visual or cognitive disability, from accessing the content.
-  **Manual Work:** The clients burden of managing, tagging and categorising such static content without a streamlined system, especially when the content spans over decades can be an insurmountable task to tackle without a helpful tool.

How this tool will help:

-  **Digitization and Search Optimization:** By converting static content and older PDF content into a digitally optimised format, readers will be able to search and locate specific content much quicker than digging through an old archive, increasing accessibility.
-  **Integrated Accessibility Tools:** This solution aims to also incorporate seamless essential accessibility tools such as zoom, adjustment of fonts, or utilising text to speech functionalities to cater to all and ensure a universally inclusive reading experience.
-  **Streamlined Admin Tool:** This concept will also aim to offer a simplified way to automate tasks such as tagging and categorising past and future content and reduce manual efforts.

---

## Required Resources & Technical Solutions

We are adopting a combination of various, popular web technologies and proven libraries. Each component of our tech stack has been selected to address a specific aspect of the problem, ensuring that the final solution is comprehensive and robust.

### Quarkus

-  **Purpose:** Easy to use API backend framework (similar to Spring) with native compilation through GraalVM with rapid development capabilities through a "dev mode" that comes with a well-maintained and diverse technology ecosystem.

- 🔧 **Usage:** Quarkus will serve as the backbone of the back end, and will allow the frontend to have a secure connection to the backend services and database. Annotation-based endpoint creation through RESTEasy will make spinning up endpoints a breeze for developers and the tech ecosystem and community behind Quarkus ensures its stability and ability to support projects of many varieties. This will be used in places where communication between the backend and frontend will need to happen
- 📐 **Transforming Design Concepts with:** An architectural Diagram and Simple UML diagrams of relevant POJOs outlining their attributes for synchronization with the Database.

### Tech to be used with Quarkus:

- Hibernate ORM with Panache ([Simplified Hibernate ORM with Panache](#)): These two will serve to translate what is in the code as business models very cleanly into the database as tables. With this there will be no need to make our own database bootstrapping, as this handles it for us and ensure that the models in the database are **always<sup>\*1</sup>** consistent with what is in the code
- JDBC connection to postgres database ([JDBC Driver - PostgreSQL](#)): This provides us an easy out-of-the-box way to connect the backend to the database with very little to zero config (Commonly known as a zero-config dev mode setup<sup>\*2</sup>). This out-of-the-box also contains a standard connection pool with [Agroal](#) in case things get dicey and we need more than one active connection to the database (unlikely for a project of this scope).
- Bryn's JUnitTesting Framework ([GitHub - Expressionless/Test-Suite: A Test Suite to make unit tests](#)): A custom JUnit framework that allows speedy and concise creation of various unit tests. This is a nice to have but will certainly speed up (and hopefully encourage) the creation of unit tests in the back end.

### Keycloak 🎨

- 🎯 **Purpose:** Out of the box authentication and authorization with secure handling of passwords.
- **Some Notes on password handling in the industry:** Password handling out in the industry is always a fragile subject, with a lot of companies (read: adobe, [Adobe 2013 data breach??](#)) implementing weak/insecure password handling that leads to easy-to-avoid vulnerabilities when it comes to user passwords.
- **How Keycloak addresses this problem and what it means for us:** Keycloak addresses this issue by storing passwords not as encrypted values, and certainly not as plaintext - but by putting passwords through an arbitrary number of hashes and salts. What this does is destructure the password from its original value and ensures that the password cannot be determined by what is produced from this process as a result ([What is hashing and how does it work?](#)). This value that is produced is what is stored.  
Keycloak will also provide us an out of the box theme-able login page. This will not be necessary for users to login to read the PDFs, however it is highly advisable that the admin page is password protected.
- 🔧 **Usage:** Frontend gets login page from keycloak. After logging in, the user will be granted an `Access Token` (AKA JsonWebToken, see: [JWT.IO](#)) which abides by the OAuth2 Standard ([JSON Web Tokens for OAuth 2.0](#)), which provides both Authentication and Authorization. Whenever the frontend is needing to provide some claim of authenticity or authority, this token can be placed in the header of any endpoint request, and it is a CDI-Injectable bean ([Contexts and Dependency Injection](#)) which provides ease of use with the backend. This will be the backbone of our security system.
- 📄 **Code Implementation & Functionality:** TODO: Implement KeycloakToken in backend

### Docker 🚀

- 🎯 **Purpose:** Provide ease of deployment through portability and self-contained environments for each service (namely: frontend, backend, minIO, keycloak, postgresql). Docker allows containerized deployment of services, and is one step before a fully-fledged Kubernetes ([Production-Grade Container Orchestration](#)) deployment. Docker will allow us to have control over which ports will be used to ensure no conflicts occur between services, by making ports seen by the service “internal” and ports seen by the server “external”. This is among one of the many wonderful things docker will provide for us, and it is an accepted standard for industry development & deployment.
- 🔧 **Usage:** Docker images will be created for the react front end, and quarkus back end. The other services (keycloak, postgresql, minIO) all have predefined images that can be downloaded with ease with configuration of a `docker-compose.yml`. This will allow easy

starting up and shutting down of any deployment on any system, providing an agile framework with which to deploy.

-  **Code Implementation & Functionality:** Will be implemented using Docker Compose ([Docker Compose overview](#)), which will allow us to configure “containers” for our Docker images to run inside. Docker Compose provides a decent level of configuration possible, and removes a lot of the messing around with things like environment variables and ports that would otherwise be necessary without Docker & Docker Compose.

## Nginx

-  **Purpose:** Provide security by allowing more fine-grained control over the requests that are made to the server. Allow a flexible and easy-to-setup configuration for multiple services.
-  **Usage:** Functionally, Nginx sits between the users using the frontend and the server. Any requests made to the server (including requests to load the website) go through nginx, and so this is a key security concept that will be used in tandem with HTTPS to provide overall security to the application.
-  **Code Implementation & Functionality:** A selection of configurations known as “sites” are available within the nginx configuration. Any particular site can have different services that are configured to listen on different ports, and Nginx can be used as a reverse-proxy to reroute traffic from a single port over many different ports, based on the url of the request.

## minIO

-  **Purpose:** Will be the place where PDFs and large text chunks are stored, with each “entity” stored against an ID and placed in a bucket.
-  **Usage:** The PDFs, associated thumbnails and other relevant media are all stored in their respective buckets, to provide organisation. The objects are not compressed in any form, providing raw data storage which is ideal for PDFs.
-  **Code Implementation & Functionality:** backend communicates with the minIO service on behalf of the frontend, where minIO is solely responsible for the storage and security of our PDFs and thumbnails. The frontend makes requests to the backend for things that are stored in minIO. Given all the layers a request must go through to reach minIO data, there is a lot of security for the data stored in the buckets in minIO.

## PostgreSQL

-  **Purpose:** Integrates the PostgreSQL database seamlessly with Java applications using the Quarkus framework, offering efficient data storage, retrieval, and management in a lightweight and reactive manner.
-  **Usage:** When combining PostgreSQL with Java and Quarkus, developers can build high-performance and scalable applications that interact with the database. Quarkus enhances Java applications by providing fast startup times and low memory footprints, making it suitable for microservices and cloud-native architectures. PostgreSQL's reliability and advanced features align well with Quarkus' capabilities, enabling developers to create data-intensive applications, RESTful APIs, and event-driven services.
-  **Code Implementation & Functionality:** To use PostgreSQL with Quarkus, developers can utilize the Hibernate ORM (Object-Relational Mapping) framework for database interactions. By defining Java entities and corresponding database tables, developers can perform CRUD (Create, Read, Update, Delete) operations seamlessly. Quarkus' extensions facilitate easy configuration of the PostgreSQL data source and integration with the application.

## Python

-  **Purpose:** Provides general purpose tools for manipulating and extracting data from PDF content.
-  **Usage:** Python is incredibly mature as a tool in the field of data science, especially when it comes to data analysis and data extraction. Popular libraries such as matplotlib, numpy and nltk can assist in solving problems related to analysing PDF document data during the problem solving stage. Additionally, libraries geared towards PDF manipulation such as PyMuPDF and pytesseract provide powerful data-extraction functionality, whilst keeping code manageable, maintainable and easily extendible.

-  **Code Implementation & Functionality:** Using Python, we can analyse and extract all the information required from PDFs for the purposes of PDF content accessibility and tagging.
- 

## Success Criteria & Technical Indicators

-  **Secure and Reliable:** Users who decide to upload their PDF content to the tool should feel confident about its reliability pertaining to the storing, serving and editing of PDF content. The storage and retrieval of PDF content, as well as any other functionality, should perform consistently in both the short and long term once the tool has been deployed.
-  **Low Maintenance Cost:** Given the lack of funding, the tool should require little funding to stay afloat post-deployment.

## Scope and Timeline

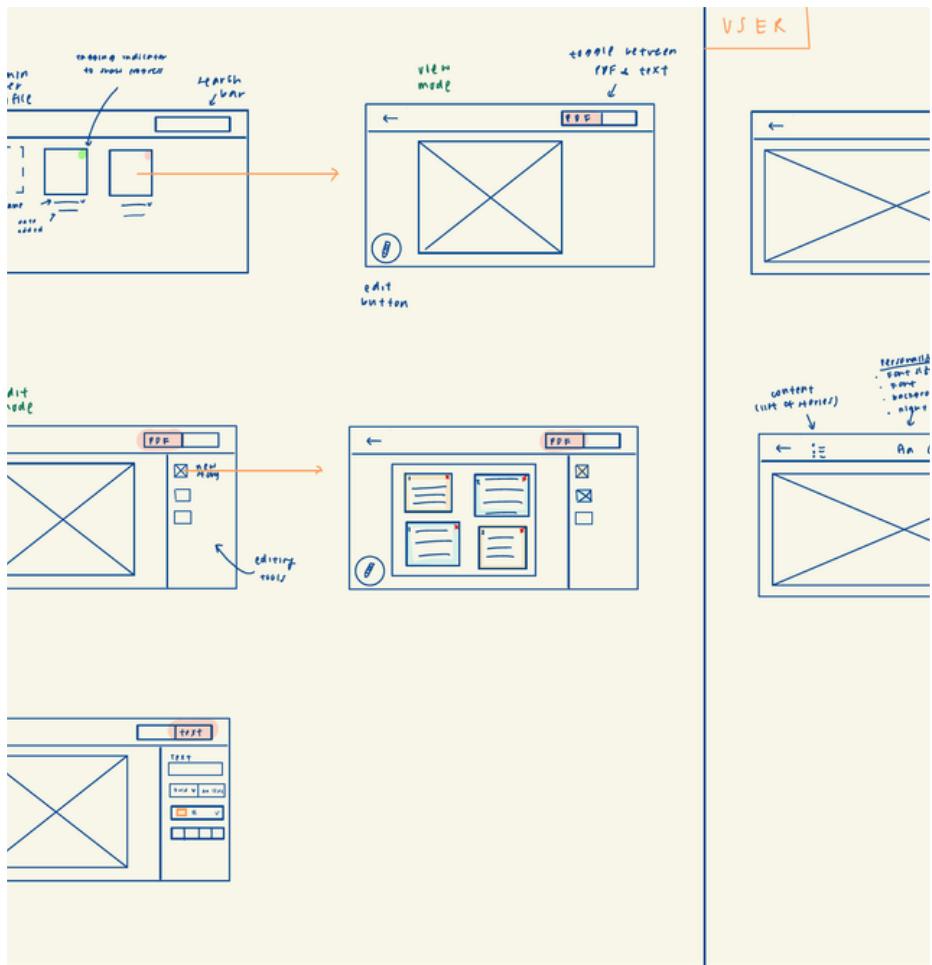
The scope of this project covers the development of two essential components to address the requirements and address issues; the pdf tool and the pdf reader tool. It's necessary that the tool be setup and functional first before the pdf reader tool can begin. It is predicted that the tool will require 1-2 weeks of initial development for the tool and another 2-3 weeks for the accessibility features and reader. This will be followed by a week of testing and feedback to ensure the application is up to standard. The final weeks will address deployment of the project.

---

Notes:

\*1: Hibernate/Panache ORM. For any Columns in the database that need to be deleted that contain existing data, first the data will need to be deleted and then the column itself can be deleted. Not an unmanageable overhead and certainly an edge case, but is something good to note.

## Initial Sketch



## Figma Design Guidelines







Nish

---

# Style Guide

---

28-08-2023

Nish

---

## Index.

01. **Typography**

02. **Colors**

03. **Iconography**

---

28-08-2023

# Typography

## Outfit

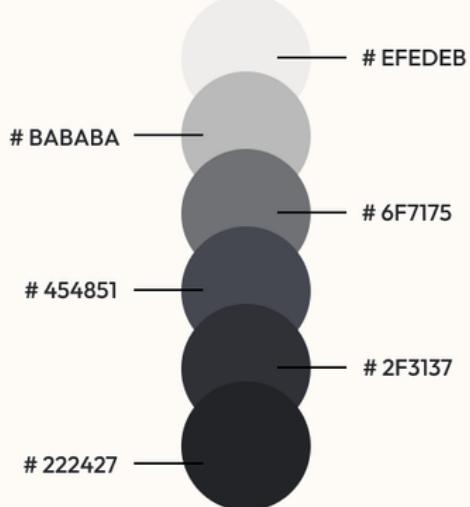
Google Fonts

Name	Type	Font size
Header	Medium	32 px
Text	Medium	24 px
Button	Regular	16 px

---

28-08-2023

# Neutral Colours



---

28-08-2023

# Primary Colours



---

28-08-2023

# Secondary Colours



---

28-08-2023

## Sprints

15 Aug 2023 - 4 Sept 2023	<a href="#"> Sprint 1</a>
5 Sept 2023 - 25 Sept 2023	<a href="#"> Sprint 2</a>
26 Sept 2023 - 16 Oct 2023	<a href="#"> Sprint 3</a>

# Sprint 1

15 Aug 2023 - 4 Sept 2023

<a href="#"> Sprint Planning 1</a>
<a href="#"> Sprint Review 1</a>
<a href="#"> Sprint Retro 1</a>

## ★ Important topics

	Topic	Goals	Results
1	Frontend	<p>Proceed on developing how the tool will handle uploading and storing of PDFs first before proceeding to the PDF editing.</p> <p>Research into text editing libraries.</p> <p>Research into WCAG accessibility guidelines, with a focus on making text accessible.</p> <p>Continue working on Figma prototypes in the meantime.</p> <p>Backend must be somewhat functional before the frontend can work on the editing functionality.</p>	<a href="#"> FIRST-63: Store PDFs in the backend through minIO <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a> <a href="#"> FIRST-60: Investigate and implement PDF to text <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a> <a href="#"> FIRST-44: Design palates and figma prototypes. <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a> <a href="#"> FIRST-20: BAU - Sprint 1 <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a>
2	Backend	<p>Set up Quarkus with Docker</p> <p>Look into how to call Python from Java</p> <p>Backend Tech stack:</p> <ul style="list-style-type: none"><li>PostgreSQL</li><li>Keycloak → MySQL</li><li>Quark</li></ul>	<a href="#"> FIRST-78: Backend Docker <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a> <a href="#"> FIRST-66: Look into minIO <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a> <a href="#"> FIRST-65: Write up proof of concept for backend. <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">DONE</span></a>

# Sprint Planning 1

<b>Date</b>	20 Aug 2023
<b>Time</b>	20:00 - 21:30
<b>Participants</b>	@Milad Izra @grace.sendjaja @Bryn Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Discord
<b>Meeting type</b>	Sprint Planning Meeting

## ★ Important topics

	<b>Topic</b>	<b>Description</b>	<b>Results</b>
1	Frontend	<p>Figure out what is feasibly easy to work on during the 2 weeks remaining in the sprint.</p> <p>Find out what requirements there are of the backends' functionality for the frontend to function as necessary.</p>	<p>Proceed on developing how the tool will handle uploading and storing of PDFs first before proceeding to the PDF editing.</p> <p>Research into text editing libraries.</p> <p>Research into WCAG accessibility guidelines, with a focus on making text accessible.</p> <p>Continue working on Figma prototypes in the meantime.</p> <p>Backend must be somewhat functional before the frontend can work on the editing functionality.</p>
2	Backend	<p>Figure out tech stack required for backend to function.</p> <p>Create backlog of tasks that should be prioritised.</p> <p>Decide on how to store the PDFs.</p>	<p>Set up Quarkus with Docker</p> <p>Look into how to call Python from Java</p> <p>Backend Tech stack:</p> <ul style="list-style-type: none"> <li>• PostgreSQL</li> <li>• Keycloak → MySQL</li> <li>• Quarkus</li> <li>• Nginx</li> <li>• Docker</li> <li>• minIO</li> </ul>

# Sprint Review 1

Date	9 Sept 2023
Time	17:00
Participants	@Milad Izra @grace.sendjaja @Bryn Meachem @matthew.thao.pham @Upnishadd
Location	Discord
Meeting type	Sprint Review

## ★ Important topics

### Review Sprint 1 Goal:

- Goal:
  - Set up an upload page and basic functionality for uploading PDFs to the backend and establishing communication between the frontend and backend. Pipeline for PDF storage.
- Outcome:
  - Frontend functionality has been established and is ahead of schedule, with the backend aiding in integrating. The backend has some more cases to work on. Backend core functionality has been established, working on the database. Overall, on track.
- Future Plan:
  - More formalised testing, Unit testing and End-to-End testing. Authentication for Admin UI.

### Developers showcase progress on goals:

- The frontend looks good :)
- The backend looks good :)

### Stakeholders discuss progress and reassess the current state of the project:

- The progress of the project looks good and on track. Continue some discussions about the features of the project and produce a finalised list of the features we can offer to the client. Discuss the plan for backups and cost analysis.

### Current Product Backlog:

- Backlog needs to be expanded for sprint 2, sprint 1 backlog is almost finished.

### Timeline and Budgets:

- Timeline: On Track.
- Cost Analysis needs to be done for: AWS (Main site and backups), NAS (Local Server), Translate Feature (Undecided), Text Extraction.

# Sprint Retro 1

## 📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the [4Ls Retrospective Play](#).

<b>Team</b>	Nish
<b>Team members</b>	@Upnishadd @Bryn Meachem @matthew.thao.pham @Bryn Meachem @grace.sendjaja
<b>Date</b>	9 Sept 2023
<b>Retrospective period</b>	Sprint 1

## 💡 4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
Nish	Enjoyed discussing with the client and also team members on what they're working on.	Less ambiguous submission documents, so I know what's missing.	Organising Confluence pages	I can do documentation tasks
Bryn	<ul style="list-style-type: none"><li>• API Work</li><li>• Microservice Work</li><li>• Teaching Backend</li></ul>	<ul style="list-style-type: none"><li>• Keycloak</li><li>• More formal discussion and coordination</li><li>• Testing</li></ul>	<ul style="list-style-type: none"><li>• Guideline writing</li></ul>	<ul style="list-style-type: none"><li>• MinIO java SDK</li><li>• Multipart file uploading</li></ul>
Matt	<ul style="list-style-type: none"><li>• Learning the backend, and how different technologies/frameworks interact with each other</li><li>• Code development on linux using vim and other development tools.</li></ul>	<ul style="list-style-type: none"><li>• Proper testing in different areas of both the front end and back end</li><li>• Figuring out how PDF data extraction is supposed to integrate with the rest of the backend/frontend.</li></ul>	<ul style="list-style-type: none"><li>• Python dependencies</li><li>• OCR text recognition</li><li>• Parsing PDF data</li><li>• Stack Overflow</li></ul>	<ul style="list-style-type: none"><li>• Quarkus</li><li>• MinIO</li><li>• PDF structure</li><li>• Java PDF libraries</li><li>• Python PDF libraries</li><li>• OCR text extraction</li><li>• Problems associated with extracting data from PDFs</li></ul>
Milad	<ul style="list-style-type: none"><li>• Working with react and associated libraries</li><li>• Building custom components &amp; hooks specifically targeting our needs</li><li>• Working with the backend to</li></ul>	N/A	Working with react-pdf & Chakra UI, Git Flow Management	Gained major confidence in react.js, libraries such as axios, react-router, redux, framer motion, redux toolkit, and especially Chakra UI

	implement features			
Grace	<ul style="list-style-type: none"> <li>Creative freedom (making colour palettes etc.)</li> <li>Also liked making client presentations.</li> </ul>	More feedback on figma about functionality and design.	Not really loathed but I feel bad for Milad when changing the designs.	Learned React from Milad + I love Latex

## ⚡ Action plan

Action	Owner	Due date	Action items
Make sure team is happy and onboard with decisions.	@Upnishadd	26 Sept 2023	Regular informal 1 on 1 contact with team members to ask about progress and thoughts of the project
Define PDF utilities more concretely through a more elaborate PDFWrapper class implementation.	@matthew.thao.pham	26 Sept 2023	Look into Java-Python communication Python Docker Unit tests for PDFWrapper Clean up Python/Java code according to guidelines
More Integration Testing. Guideline Enforcement	@Bryn Meachem	26 Sept 2023	<ul style="list-style-type: none"> <li>Look into Mockito</li> <li>Write Quarkus Endpoint Tests</li> </ul> Guideline related: <ul style="list-style-type: none"> <li>Focus on Code Smell for a couple days</li> <li>Discuss with @matthew.thao.pham about enforcing guidelines</li> </ul>
Work on smaller feature branches & More testing for existing features	@Milad Izra	26 Sept 2023	<ul style="list-style-type: none"> <li>Start the new feature branch next week <code>feat-pdf-zoom</code> to implement zooming capabilities [Smaller Feature Branches]</li> <li>Create new Unit Tests &amp; Integration Tests next week.</li> </ul>
Reach out to members more often to get feedback and be on the same page.  Be more involved in the code	@grace.sendjaja	26 Sept 2023	Add shadows and other details to figma then communicate with teammates  Make the custom text editor with quill

# Sprint 2

5 Sept 2023 - 25 Sept 2023

<a href="#"> Sprint Planning 2</a>
<a href="#"> Sprint Review 2</a>
<a href="#"> Sprint Retro 2</a>

## ⭐ Important topics

	Topic	Goals	Results
1	PDF Text Extraction Tool	With the uploading done, we can move to fleshing out the core functionality of the PDF tool, which is the PDF data extraction. Come up with a list of tasks that we can put onto our product backlog to proceed working on.	<a href="#"> FIRST-34: While in Edit mode, viewing the PDF form of the file, Admins can create/edit/manage stories that are linked to some existing text. <span style="border: 1px solid green; padding: 2px;">DONE</span></a>
2	PDF Text Edit Tool	Given the PDF data extraction is proceeding with development, we need to figure out how to display and edit text on the front end, both in terms of implementation details and design.	<a href="#"> FIRST-35: While in Edit mode, viewing the text form of the file, Admins can edit/review the story content and change the appearance of text <span style="border: 1px solid green; padding: 2px;">DONE</span></a>
3	PDF Modelling and Storage	With the Admin Edit page being worked on, we need to figure out what information is necessary to be stored in the backend, and how we're serializing both: 1. Data generated in the frontend 2. Data uploaded to the backend  This requires finding time to define the model's we're constructing in our PostgreSQL database.	<a href="#"> FIRST-30: Admins can add new PDFs they wish to store/edit <span style="border: 1px solid green; padding: 2px;">DONE</span></a> <a href="#"> FIRST-95: Save the document with all related changes <span style="border: 1px solid green; padding: 2px;">DONE</span></a> <a href="#"> FIRST-30: Admins can add new PDFs they wish to store/edit <span style="border: 1px solid green; padding: 2px;">DONE</span></a>

## Sprint Planning 2

Date	30 Sept 2023
Time	23:56
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Discord
Meeting type	Sprint Planning Meeting

### ⭐ Important topics

# Sprint Review 2

Date	30 Sept 2023
Time	23:10
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Discord
Meeting type	Sprint Review

## ★ Important topics

### Review Sprint 2 Goal:

- Goal:
  - Integrate backend and frontend properly, with features such as text extraction and text viewing being functional
  - Proper persistence of data in the backend over multiple sessions
- Outcome:
  - A little behind on schedule simply due to assignment hell.
  - Backend needs some more time on properly storing and persisting data
  - Frontend is bottlenecked when it comes to text extraction functionality due to lack of endpoints in the backend
  - Backend however is set up to speed up development going into Sprint 3. Same with the frontend
  - Overall, we're behind but in a good position to get features finished by Sprint 3, given enough work.
- Future Plan:
  - Increased communication between frontend and backend when things go awry.
  - Proper E2E testing.
  - Authentication.
  - User Site.

### Developers showcase progress on goals:

- The frontend looks good :)
- The backend looks good :)

### Stakeholders discuss progress and reassess the current state of the project:

- Given we have 3 weeks remaining, we may have to cut some non-critical features. Nonetheless, we have the opportunity to keep progress on track to hopefully present a finished product to the client at the end of Sprint 3.

### Current Product Backlog:

- There are some out-standing tasks that will have to be carried over from Sprint 2 into Sprint 3. Nevertheless, we will start to create more tasks in Sprint 3 related to both the User View and testing.

### Timeline and Budgets:

- Timeline: Slightly behind, but close to on track.

- Possible options for discussing the hand-over of the product to the client.

# Sprint Retro 2

## 📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the 4Ls Retrospective Play.

Team	Nish
Team members	@Upnishadd @Bryn Meachem @matthew.thao.pham @Bryn Meachem @grace.sendjaja
Date	30 Sept 2023
Retrospective period	Sprint 2

## 💡 4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
Milad	<ul style="list-style-type: none"><li>Working With React</li></ul>	<ul style="list-style-type: none"><li>More time</li></ul>	<ul style="list-style-type: none"><li>Time Commitment</li></ul>	<ul style="list-style-type: none"><li>Worked with React Redux Thunk &amp; Asynchronous</li></ul>
Nish	Project starting to come together now 😊	Abit more time to finish the project	Scope of Project as time crunch approaches with other Assignments	Confluence Cleanup is more than just structured ordering and there's a lot more requirements, cat pics wont get you 15/15
Bryn	<ul style="list-style-type: none"><li>Working with Database models/ORM</li></ul>	<ul style="list-style-type: none"><li>More time</li><li>A better handle over my JIRA tickets</li></ul>	<ul style="list-style-type: none"><li>Code documentation</li><li>Assignments and other commitments</li></ul>	<ul style="list-style-type: none"><li>Feature creep is annoying</li><li>Active Record vs DAO vs Repository Pattern is annoying decision to make</li></ul>
Matt	<ul style="list-style-type: none"><li>The first time I ran python through Java I cried with joy.</li><li>In fact I cried double tears of joy.</li><li>Call it negative sadness</li></ul>	<ul style="list-style-type: none"><li>Time.</li></ul>	<ul style="list-style-type: none"><li>Balancing IT project and other uni assignments</li></ul>	<ul style="list-style-type: none"><li>White space in Java is not the same as white space in python.</li><li>Endpoints</li><li>Panache</li></ul>
Grace	Being involved in the code and all the helpful guidance from teammates ❤️	wished i hadn't made the figma look like that	Assignments and other commitments	<ul style="list-style-type: none"><li>Quill &amp; React</li></ul>

## ⚡ Action plan

Action	Owner	Due date	Action items
--------	-------	----------	--------------

Help organise, wrapping up the project	@Upnishadd	16 Oct 2023	Contact team members, arrange and organise what needs to be done from each side of things, and bring together the different parts.
Communicate and collaborate with frontend more proactively.	@matthew.thao.pham	16 Oct 2023	Define more concretely what is required of the backend from the frontend and finish those tasks in a timely fashion.
Check in with Frontend more often	@Bryn Meachem	16 Oct 2023	<ul style="list-style-type: none"> <li>• Ensure that issues in the backend that are picked up by the frontend are not roadblocking the frontend</li> </ul>
Set aside time to polish up the backend	@Bryn Meachem @matthew.thao.pham	16 Oct 2023	<ul style="list-style-type: none"> <li>• Integrate Keycloak</li> <li>• Do SSL</li> <li>• Finish Endpoints</li> </ul>
Help out in Frontend	@grace.sendjaja	16 Oct 2023	Contact Milad, discuss which parts need to be completed and start testing and defect fixing.
Finish Up Frontend Tasks	@Milad Izra	16 Oct 2023	Contact Nish to finalise all remaining tasks and finish off the frontend.

## Sprint 3

26 Sept 2023 - 16 Oct 2023

<a href="#"> Sprint Planning 3</a>
<a href="#"> Sprint Review 3</a>
<a href="#"> Sprint Retro 3</a>

# Sprint Planning 3

Date	30 Sept 2023
Time	18:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Discord
Meeting type	Sprint Planning Meeting

## ★ Important topics

	Topic	Description	Results
1	E2E functionality	Get text extraction and searching completed, and properly functioning in the context of the Edit Page and the PDF Management Page.	Carry over tasks from Sprint 2 that have to be completed. Flesh out more tasks to quickly resolve any current issues. Consult @Milad.Izra and @grace.sendjaja for more info on how to develop the functionality of the backend to assist with development of the frontend.  [ ] SEC-7: Stories Endpoints <span style="border: 1px solid green; padding: 2px;">DONE</span>
2	Authorisation	We need authy for admins who would like to upload PDFs. Add ability for client to add more admins.	Get Keycloak react module on front end and Keycloak service running on the backend. Get frontend to communicate with Keycloak service using react module. Get it up and going.  [ ] FIRST-70: Backend Bootstrap <span style="border: 1px solid green; padding: 2px;">DONE</span>
3	User Site/ User View	Users need to be able to access PDFs to read. Create a user site to allow for external viewing of PDFs/Editions	Communicate with @Milad.Izra on routing to create a separate "site" for Users, or other alternatives. Ideally, the functionality of the User Page should be similar or the same as the Edit Page.  [ ] FIRST-28: Allow external users to view PDFs using external link <span style="border: 1px solid green; padding: 2px;">DONE</span>  [ ] SEC-14: The Reader Page <span style="border: 1px solid green; padding: 2px;">DONE</span>

# Sprint Review 3

Date	21 Oct 2023
Time	21:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Discord
Meeting type	Sprint Review

## ★ Important topics

### Review Sprint 3 Goal:

- Goal:
  - Finalise Text Extraction, Finalise Rich Text Editor Functionality, Authentication and User View.
- Outcome:
  - Spoke to the client about reducing the scope and was agreed upon.
  - Text extraction is working
  - Text document is saving in backend and is retrievable
  - Keycloak essentially completed just doing touchups
  - Backend and Frontend are sorted and finalising touchups for client handover
  - Overall, Solid achievement from the team
  - Might need a bit more time to touch on authentication
- Future Plan:
  - Handover.
  - Touchups.
  - Submissions.

### Developers showcase progress on goals:

- The frontend looks good :)
- The backend looks good :)

### Stakeholders discuss progress and reassess the current state of the project:

- Progress looks good, the project is in a good place and we are ready to hand it over with a few touchups.

### Current Product Backlog:

- No real backlog left :).

### Timeline and Budgets:

- Timeline: On track.

# Sprint Retro 3

## 📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the [4Ls Retrospective Play](#).

<b>Team</b>	Nish
<b>Team members</b>	@Upnishadd @Bryn Meachem @matthew.thao.pham @Bryn Meachem @grace.sendjaja
<b>Date</b>	21 Oct 2023
<b>Retrospective period</b>	Sprint 3

## 💡 4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
@Upnishadd	<ul style="list-style-type: none"><li>Doing some Dev work.</li></ul>	<ul style="list-style-type: none"><li>Less reports</li></ul>	<ul style="list-style-type: none"><li>Reports</li></ul>	<ul style="list-style-type: none"><li>Importance of having more buffers and time constraint discussions.</li><li>Smaller Project can be better.</li></ul>
@Milad Izra	<ul style="list-style-type: none"><li>Seeing the project come together</li></ul>	<ul style="list-style-type: none"><li>More Time</li></ul>	<ul style="list-style-type: none"><li>Time Commitment</li></ul>	<ul style="list-style-type: none"><li>Efficient Code Implementation For API Calls</li></ul>
@Bryn Meachem	<ul style="list-style-type: none"><li>Presenting the project</li><li>Polishing the code</li></ul>	<ul style="list-style-type: none"><li>Less assignments</li></ul>	<ul style="list-style-type: none"><li>There isn't really anything I loathed this time with regards to the team or the project</li></ul>	<ul style="list-style-type: none"><li>Teamwork skills</li></ul>
@matthew.thao.pham	<ul style="list-style-type: none"><li>Writing up all the endpoints</li><li>Seeing the work in the backend slowly get integrated with the frontend</li></ul>	<ul style="list-style-type: none"><li>Better understanding of DB</li><li>Perhaps have a chance to cover some Devops</li></ul>	<ul style="list-style-type: none"><li>Reports</li></ul>	<ul style="list-style-type: none"><li>Learned the importance of just getting things working and not worrying too much about having the cleanest code</li></ul>

<b>⚡ Action plan</b>	@grace.sendjaja	<ul style="list-style-type: none"> <li>• Learning about endpoints</li> <li>• Connecting the</li> </ul>	<ul style="list-style-type: none"> <li>• More time and less clashes with other</li> </ul>	<ul style="list-style-type: none"> <li>• Reports</li> </ul>	<ul style="list-style-type: none"> <li>• Learned that Git is not as scary as it looks</li> </ul>
	Action	Owner	Due date	Action items	
Finalise Documentation	@Upnishadd		12 Nov 2023	Clear up any documentation needed to be done.	
Clean Up Frontend Codebase	@Milad Izra		12 Nov 2023	Clean up codebase, and refactor existing code	
Document Backend	@Bryn Meachem		12 Nov 2023	Document Backend, ensure Google Style Guide compliant. Write a couple integration tests	
Support Frontend during API integration phase	@matthew.thao.pham		12 Nov 2023	Be on standby and fix potential bugs that arise during testing	
Connect Endpoints	@grace.sendjaja		12 Nov 2023	Communicate with backend to finalise the quill page	

## Meetings Page

<a href="#"> Weekly Standup Meetings</a>
<a href="#"> Client Meetings</a>

## Weekly Standup Meetings

Meeting Link
<a href="#"> Weekly Meeting 1</a>
<a href="#"> Weekly Meeting 2</a>
<a href="#"> Weekly Meeting 3</a>
<a href="#"> Weekly Meeting 4</a>
<a href="#"> Weekly Meeting 5</a>
<a href="#"> Weekly Meeting 6</a>
<a href="#"> Weekly Meeting 7</a>
<a href="#"> Weekly Meeting 8</a>
<a href="#"> Weekly Meeting 9</a>
<a href="#"> Weekly Meeting 10</a>
<a href="#"> Weekly Meeting 11</a>

# Weekly Meeting 1

Date	26 Jul 2023
Time	11:00 - 13:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	PAR-Alan Gilbert-123
Meeting type	Weekly Meeting
Materials	

## ⭐ Important topics

	Topic	Description	Results
1	Pages	Confluence, Jira, Git, Discord pages needed to be setup	<p>The pages were set up with the following links:</p> <p>Discord: <a href="#">Join the IT Project Discord Server!</a></p> <p>Git: <a href="https://gitlab.eng.unimelb.edu.au/nishgangan">https://gitlab.eng.unimelb.edu.au/nishgangan</a></p> <p>Jira:</p> <div style="border: 1px solid #ccc; padding: 10px; width: fit-content;"><p><b>Show your content</b></p><p>This may be because you're not logged in. Try opening the issue in Jira.</p></div>

2	Team Registration	Team needed to register to a workshop	Registered to Planimation workshop, requested a more tech-focused project
---	-------------------	---------------------------------------	--

## Weekly Meeting 2

Date	2 Aug 2023
Time	11:00 - 13:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	PAR-Alan Gilbert-123
Meeting type	Weekly Meeting
Materials	

### ★ Important topics

	Topic	Description	Results
1	Switch Workshop	Were informed Planimation would be research based project, so we need to arrange to switch to a different workshop for a different project	Arranged with @Ben.Evans to join his workshop and work on Flemington Kensington News.
2	Tech Stack	Figure out which Tech stack we will be using	Front End: React Database: Postgres Backend: Quarkus DevOps: Docker Deployment Authentication: Keycloak

# Weekly Meeting 3

<b>Date</b>	8 Aug 2023
<b>Time</b>	09:00 - 11:00
<b>Participants</b>	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Cussonia Court, Room 2
<b>Meeting type</b>	Weekly Meeting
<b>Materials</b>	

## ★ Standup

	<b>Team Member</b>	<b>Update</b>
1	Nish	Worked with the team on organising some communication platforms such as Jira and reached out to the Client for a quick greetings, plan to grasp a stronger understanding of the project this week. No blockers.
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Jira setup</li> <li>• Confluence setup</li> <li>• Github setup</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• User stories/reqs</li> <li>• Project decision</li> </ul>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Jira setup</li> <li>• Confluence setup</li> <li>• Github setup</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• User stories/reqs</li> <li>• Project decision</li> </ul>
4	Milad	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• I looked into the various project ideas and the requirements. I saw that squarespace was being used for the website and begun researching into various avenues into implementing the tools.</li> <li>• Jira Setup</li> </ul>

		<ul style="list-style-type: none"> <li>• Confluence Setup</li> <li>• Git Setup</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Project Descision on direction</li> <li>• Project ideas</li> </ul>
5	Matt	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Jira setup</li> <li>• Confluence setup</li> <li>• Github setup</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• User stories/reqs</li> <li>• Project decision</li> </ul>

# Weekly Meeting 4

<b>Date</b>	15 Aug 2023
<b>Time</b>	09:00 - 11:00
<b>Participants</b>	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Cussonia Court, Room 2
<b>Meeting type</b>	Weekly Meeting
<b>Materials</b>	

## ★ Standup

	<b>Team Member</b>	<b>Update</b>
1	Nish	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Contacted Client to gather some more information</li> <li>• Drafted and Sent over a proposal to the client</li> <li>• Aid in design formation</li> </ul> <p>Today</p> <ul style="list-style-type: none"> <li>• JIRA Backlog</li> <li>• Finalise formulating design</li> </ul> <p>Problems</p> <ul style="list-style-type: none"> <li>• No real problems as the Product Owner, but will help other team members with their tasks</li> </ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Coordinate with the team to decide more on tech stack</li> <li>• Screens design and sketch out roadmap</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• the weeds (getting into the details too much)</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Setting up a JIRA backlog</li> <li>• Getting wireframes going for screens and potentially doing something with Figma or otherwise</li> <li>• Sending vesta the quarkus quick start</li> </ul> <p>Getting my foot checked out :(</p>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Long meeting about design and functionalities of tool</li> <li>• Listened in on backend discussions</li> <li>• Made presentation to help client visualise idea during pitch</li> </ul> <p>Problems:</p>

		<ul style="list-style-type: none"> <li>• None</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Finish wireframe of design</li> </ul>
4	Milad	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Super long meeting where we discussed the frontend. We agreed on a 'design' (kinda) and a conceptual design</li> <li>• Research to address a problem we had</li> <li>• Looked into PDF.js and PDF-LIB.js</li> <li>• Insane discovery of OCR</li> </ul> <p>Today</p> <ul style="list-style-type: none"> <li>• JIRA Backlog</li> <li>• Finalise Conceptual Design (Hopefully) and start implementation of the front end</li> </ul> <p>Problems</p> <ul style="list-style-type: none"> <li>• How to go about implementing some of the things we discussed. However looking into the some of the .js libraries may address this problem. More to come soon</li> </ul>
5	Matt	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Meeting to solidify (sort of) functional requirements and design of the product</li> <li>• Investigated pros and cons of different PDF parsing methods, and starting to investigate OCR through Tesseract OCR</li> </ul> <p>Today</p> <ul style="list-style-type: none"> <li>• JIRA Backlog</li> <li>• Finalise Conceptual Design and start implementation of the front end. Solidify team roles and get a working process going, and solidify tech stack.</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• How does one Scrum Master</li> </ul>

# Weekly Meeting 5

Date	22 Aug 2023
Time	09:00 - 11:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Cussonia Court, Room 2
Meeting type	Weekly Meeting
Materials	

## ★ Standup

	Team Member	Update
1	Nish	<p>Yesterday</p> <ul style="list-style-type: none"><li>• Worked on Confluence Pages</li><li>• Worked on Python Script to clean text</li><li>• Caught up with team members</li><li>• Meetings</li></ul> <p>Today</p> <ul style="list-style-type: none"><li>• Meetings</li><li>• More Confluence</li><li>• Submission Document</li></ul> <p>Problems</p> <ul style="list-style-type: none"><li>• None</li></ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"><li>• Begun setting up docker images for backend, keycloak, minio</li><li>• Started writing the Backend Proof of Concept</li><li>• Caught up with team members</li><li>• Meetings</li></ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"><li>• Meetings</li><li>• Finish off the Backend Proof of Concept + Architectural Diagram</li><li>• Setup AWS EC2, initial deployment with an NGINX configuration</li><li>• Backend Sprint Planning</li></ul>

		<ul style="list-style-type: none"> <li>• Finish off MinIO configuration and deploy to EC2 in a follow up deployment for the frontend</li> </ul>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Finishing up UI mockups</li> <li>• Joined in meetings yay</li> <li>• Doing the project assessment submission</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• None</li> </ul> <p>Today + rest of week:</p> <ul style="list-style-type: none"> <li>• Read about the guidelines for accessibility to be understood by screen reader</li> </ul>
4	Milad	
5	Matt	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Continued to work on and finish PDF text extraction</li> <li>• Did some sprint planning for the front end</li> <li>• Caught up with team members</li> <li>• Meetings</li> </ul> <p>Today</p> <ul style="list-style-type: none"> <li>• Meetings</li> <li>• More Confluence</li> <li>• Backend Sprint Planning (At least at the time)</li> </ul> <p>Problems</p> <ul style="list-style-type: none"> <li>• None</li> </ul>

# Weekly Meeting 6

Date	29 Aug 2023
Time	09:00 - 11:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Cussonia Court, Room 2
Meeting type	Weekly Meeting
Materials	

## ★ Standup

	Team Member	Update
1	Nish	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Documentation - Final Submissions</li> <li>• Confluence Prettying up</li> <li>• Arranged and held meeting with client</li> </ul> <p>Problems: <a href="#">[]</a></p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Continue catching up with team members</li> <li>• Assist team members where needed</li> <li>• Informal meetings on the nightly</li> <li>• Continue Polishing Documentation/BRD + Send BRD to client if ready</li> </ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Architectural Diagram + POC for Backend</li> <li>• Nginx Configuration for EC2</li> <li>• Dockerized Backend, Minio, Frontend, Keycloak, Postgresql</li> <li>• Found PDF Upload POC for Quarkus RESTEasy, To be used as an example for backend implementation</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• Nginx configuration is dicey</li> <li>• Quarkus RESTEasy is not documented enough</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Version Control Design Discussion</li> <li>• Integrate PDF Uploader</li> <li>• Work on postgres models</li> </ul>

3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Worked on figma</li> <li>• Finished Style guides</li> <li>• Worked on LaTeX submission</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• Merge conflict with styling ??</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Finish the visual styling</li> <li>• Upload Figma to confluence</li> </ul>
4	Milad	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Created PDF Viewer, with buffered loading.</li> <li>• Uploading of PDFs</li> <li>• React Routing</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• Version Control (Needs a Design Discussion)</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Integrate Frontend with Backend</li> <li>• Uploading Testing</li> <li>• Integration Testing</li> <li>• Bug fixing</li> </ul>
5	Matt	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Worked on PDF extraction script, making it more usable</li> <li>• Bug fixes</li> <li>• Worked through POCs + Architectural Diagrams</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• Unsure on the input for the python scripts, a few solutions at hand</li> <li>• Tag extraction (Discussion with @Nish), how the tag extraction is going to look on the editor</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Documentation</li> <li>• Assist team members where necessary</li> <li>• Sprint backlog review, go from there</li> </ul>

# Weekly Meeting 7

<b>Date</b>	5 Sept 2023
<b>Time</b>	09:00 - 11:00
<b>Participants</b>	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Cussonia Court, Room 2
<b>Meeting type</b>	Weekly Meeting
<b>Materials</b>	

## ★ Standup

	<b>Team Member</b>	<b>Update</b>
1	Nish	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Documentation - Final Submissions</li> <li>• Confluence Prettying up</li> <li>• Meetings with team</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Continue catching up with team members</li> <li>• Assist team members where needed</li> <li>• Informal meetings on the nightly</li> <li>• Sprint Planning, Sprint Review, Sprint Retro</li> </ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Kill old EC2</li> <li>• Start new EC2 with more RAM</li> <li>• Upload Endpoint</li> <li>• Thumbnail Endpoint</li> <li>• MinIO integration with backend</li> <li>• Guidelines Writing</li> <li>• Nginx config</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• There are a lot of API guidelines</li> <li>• react-redux not behaving within docker (config issue in Dockerfile)</li> </ul> <p>This Week:</p> <ul style="list-style-type: none"> <li>• Keycloak Integration</li> </ul>

		<ul style="list-style-type: none"> <li>• Postgres Integration</li> <li>• SSL/Letsencrypt into Nginx</li> <li>• Further endpoints for Frontend as needed</li> <li>• Spot instance</li> </ul>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Meetings with team</li> <li>• Starting on pdf edit page with quill</li> </ul> <p>Problems: [idk how to and need to learn how to use quill]</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Continue catching up with team members</li> <li>• Informal meetings on the nightly</li> <li>• Sprint Planning, Sprint Review, Sprint Retro</li> </ul>
4	Milad	<p>Last Week:</p> <ul style="list-style-type: none"> <li>• Added Icons to Header</li> <li>• Thumbnail Sidebar is Functional</li> <li>• View Page Sidebar animations and works</li> <li>• Unit Tests and Integration Tests</li> <li>• Improved Codebase</li> </ul> <p>Problems: []</p> <p>TODO:</p> <ul style="list-style-type: none"> <li>• Finish View Page</li> <li>• Test DocumentLoader</li> </ul>
5	Matt	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Meetings with team</li> <li>• Learning Sly's PDF Super Backend Alleyway Highway</li> <li>• Looking into PDF tooling for Java, including software structure and performance</li> <li>• Generating and uploading thumbnails</li> <li>• Cleaning up documentation</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• There's a lot to learn when it comes to both Java and every framework Sly has decided to use</li> <li>• OCR is still something that needs to be addressed</li> <li>• How we're dockerizing python and dealing with python dependencies</li> <li>• Testing is primitive at best, quite tedious to confirm that code is functioning correctly</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Continue catching up with team members</li> </ul>

- Informal meetings on the nightly
- Sprint Planning, Sprint Review, Sprint Retro
- Investigating into OCR once more

# Weekly Meeting 8

<b>Date</b>	12 Sept 2023
<b>Time</b>	09:00 - 11:00
<b>Participants</b>	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Cussonia Court, Room 2
<b>Meeting type</b>	Weekly Meeting
<b>Materials</b>	

## ★ Standup

	<b>Team Member</b>	<b>Update</b>
1	Nish	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Continue catching up with team members</li> <li>• Assist team members where needed</li> <li>• Informal meetings on the nightly</li> <li>• Continue Polishing Documentation/BRD + Send BRD to client if ready</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Documentations</li> <li>• Confluence Prettying up</li> <li>• Meetings with Backend, for AWS and other models</li> </ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Sprint Planning</li> <li>• Investigate Spot Instance</li> <li>• Planning postgres models</li> <li>• Postgres Integration</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Develop models for DB with Vesta</li> <li>• Further keycloak/SSL integration</li> </ul>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Edit text page looks ok and is functional :&gt;</li> <li>• learning from chef milad</li> <li>• Redesigning to compromise with quill idk yet</li> </ul>

		<p>Problems: [i hate week 3-4 grace who made figma]</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• continue experimenting</li> <li>• also idk how the backend is going to work for this page</li> </ul>
4	Milad	<p>Yesterday:</p> <p>Integrated Backend into Frontend</p> <p>Bug Fixes</p> <p>Testing</p> <p>Drawing Boxes Investigation</p> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Begin Drawing Boxes Feature</li> <li>• Test and coordinate with backend on coordinates system</li> </ul>
5	Matt	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Sprint Planning</li> <li>• Returned back to python OCR, familiarising myself with libraries</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• Possible issues with processing power of server</li> <li>• Python library dependencies are awkward</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Continue with OCR work</li> <li>• Develop models for database</li> <li>• Develop PDFWrapper class further</li> </ul>

# Weekly Meeting 9

Date	19 Sept 2023
Time	09:00 - 11:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Cussonia Court, Room 2
Meeting type	Weekly Meeting
Materials	

## ★ Standup

	Team Member	Update
1	Nish	<p>Yesterday:</p> <ul style="list-style-type: none"><li>• Documentations</li><li>• Confluence Prettying up</li><li>• Meetings with Backend, for AWS and other models</li></ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"><li>• Documentations</li><li>• Confluence Prettying up</li><li>• Meetings with Backend.</li><li>• Progress Check 2.</li></ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"><li>• Database Models</li><li>• Text Extraction Pipeline Discussion</li><li>• Documentataion Review</li></ul> <p>Problemms: []</p> <p>Today:</p> <ul style="list-style-type: none"><li>• Database Resources (not Models)</li><li>• Impl java-side text extraction</li><li>• SSL</li></ul>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"><li>• Looking for inspo to style edit page</li></ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"><li>• Meet with Sly about backend</li></ul>

		<ul style="list-style-type: none"> <li>• Style edit page</li> </ul>
4	Milad	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Feat-draw-box has been made and added</li> <li>• Coordinated with Vesta to create the coordinates system [in progress]</li> </ul> <p>Problems: [ Coordinates System ]</p> <p>Today</p> <ul style="list-style-type: none"> <li>• Figure out coordinates system with Vesta &amp; Bug Fixes</li> <li>• Start on Auth this week</li> </ul>
5	Matt	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Able to run python scripts in java</li> <li>• Python dependencies added to Docker</li> <li>• PDFWrapper can now scan text from pdf documents and serialize the information as a JSON object</li> </ul> <p>Problems : [ Coordinates System ]</p> <p>Today</p> <ul style="list-style-type: none"> <li>• Figure out coordinates system with Milad &amp; Code Commenting</li> <li>• Elaborate on text extraction</li> <li>• Start on Auth</li> </ul>

# Weekly Meeting 10

<b>Date</b>	3 Oct 2023
<b>Time</b>	09:00 - 11:00
<b>Participants</b>	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Cussonia Court, Room 2
<b>Meeting type</b>	Weekly Meeting
<b>Materials</b>	

## ★ Standup

	<b>Team Member</b>	<b>Update</b>
1	Nish	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Documentations</li> <li>• Confluence Prettying up</li> <li>• Meetings with Backend.</li> <li>• Progress Check 2.</li> </ul> <p>Problems: <a href="#">[]</a></p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Documentations</li> <li>• Confluence Prettying up</li> <li>• Meetings with Backend.</li> <li>• Progress Check 3.</li> </ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Database Resources (not Models)</li> <li>• Impl java-side text extraction</li> <li>• SSL</li> </ul> <p>Problems: <a href="#">[]</a></p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Wrapping up reports</li> <li>• Presentation Design</li> <li>• Search Endpoint</li> <li>• Discussions with Matt and Milad about Frontend Scope</li> </ul>
3	Grace	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Styling of the edit page</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• [i don't know how endpoints work]</li> </ul> <p>Today:</p>

		<ul style="list-style-type: none"> <li>• Discuss w/ Vesta regarding endpoints</li> </ul>
4	Milad	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Added ordering to rectangles &amp; Basic Reader</li> </ul> <p>Problems:</p> <ul style="list-style-type: none"> <li>• How Implement Backend For Stories Into Frontend???</li> </ul> <p>Today:</p> <ul style="list-style-type: none"> <li>• Implement API into frontend</li> <li>• Discuss w/ Vesta Regarding API</li> </ul>
5	Matt	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Not much 😊</li> </ul> <p>Problems :</p> <ul style="list-style-type: none"> <li>• Confused on direction of project</li> <li>• Having trouble progressing since unsure of what frontend needs to function</li> </ul> <p>Today</p> <ul style="list-style-type: none"> <li>• Sort out Jira tickets</li> <li>• Draft a roadmap for the rest of remaining tasks in the project</li> <li>• Work on endpoints</li> </ul>

# Weekly Meeting 11

<b>Date</b>	10 Oct 2023
<b>Time</b>	09:00 - 11:00
<b>Participants</b>	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
<b>Location</b>	Cussonia Court, Room 2
<b>Meeting type</b>	Weekly Meeting
<b>Materials</b>	

## ★ Standup

	<b>Team Member</b>	<b>Update</b>
1	Nish	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Documentations</li> <li>• Confluence Prettying up</li> <li>• Meetings with Backend.</li> <li>• Progress Check 3.</li> </ul> <p>Problems: <a href="#">[]</a></p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Documentations</li> <li>• Confluence Prettying up</li> <li>• Meetings with Backend.</li> <li>• Progress Check 3.</li> <li>• Ethics Report</li> <li>• Presentation</li> </ul>
2	Bryn	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Wrapping up reports</li> <li>• Presentation Design</li> <li>• Search Endpoint</li> <li>• Discussions with Matt and Milad about Frontend Scope</li> </ul> <p>Problems: <a href="#">[]</a></p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Assist with endpoint design for the remaining endpoints</li> <li>• Ensure the server is ready for hand over</li> <li>• Keycloak Integration</li> </ul>
3	Grace	<p>Yesterday:</p> <ul style="list-style-type: none"> <li>• Implement endpoints in edit page (fetching and updating)</li> <li>• Help with presentation powerpoint</li> <li>• Progress check 3</li> </ul>

		<ul style="list-style-type: none"> <li>• Tidying confluence</li> </ul> <p>Problems: []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Finalize progress check 3</li> </ul>
4	Milad	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Added zoom functionality &amp; Scaling PDF's with rectangles. [Max Scale of 1.2, Min 0.6]</li> <li>• Implemented new endpoints</li> <li>• Fixed a bug regarding useEffect triggering infinite number of API calls to `/v2/story/list/\${id}`</li> <li>• Fixed a bug regarding last rectangle on canvas not being deleted</li> <li>• Fixed a bug regarding react-pdf-worker failing to work on build / prod version</li> </ul> <p>Problems : []</p> <p>Today:</p> <ul style="list-style-type: none"> <li>• Clean up code</li> <li>• Zoom Functionality</li> </ul>
5	Matt	<p>Yesterday</p> <ul style="list-style-type: none"> <li>• Wrote up majority of endpoints needed for the service to function</li> <li>• Cleared all the Jira tickets relating to necessary backend tasks</li> </ul> <p>Problems :</p> <ul style="list-style-type: none"> <li>• None</li> </ul> <p>Today</p> <ul style="list-style-type: none"> <li>• Keywords</li> <li>• Integrating API with frontend</li> <li>• Deploy new backend endpoints</li> </ul>

## Client Meetings

<a href="#"> Client Meeting 1</a>
<a href="#"> Client Meeting 2</a>
<a href="#"> Client Meeting 3</a>

# Client Meeting 1

Date	9 Aug 2023
Time	15:30 - 16:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Zoom
Meeting type	Client Meeting
Materials	 <a href="#">First client meeting</a>

## ★ Important topics

	Topic	Description	Results
1	Meet with client	Meet with client Present an initial proposal Get Client feedback	Client was happy with our presentation, and asked for a written proposal which was arranged.  <a href="https://www.youtube.com/watch?v=rKbsKGeWqHU">https://www.youtube.com/watch?v=rKbsKGeWqHU</a>

## Client Meeting 2

Date	29 Aug 2023
Time	09:00 - 11:00
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd @Ben.Evans
Location	Zoom
Meeting type	Client Meeting
Materials	<a href="#">Client Meeting #2</a>

### ★ Important topics

	Topic	Description	Results
1	Website Design	Present the planned website design.	Client was happy with the design
2	Prototype Website	Present a prototype website for the client to be able to play around with.	Client was happy with the prototype
3	Project Overview	Provide a project overview to the client.	Client was happy with the overview, had a question about one potential feature.





## Client Meeting 3

Date	5 Sept 2023
Time	16:00 - 16:30
Participants	@Milad.Izra @grace.sendjaja @Bryn.Meachem @matthew.thao.pham @Upnishadd
Location	Zoom
Meeting type	Client Meeting
Materials	

### ★ Important topics

	Topic	Description	Results
1	Reduce Scope of Project	Meet with client to explain reduction of scope for features	Client was happy with reduction in scope - sent email to @Ben Evans
2	Showcase of Project	Show what we have so far in terms of the project	Client was happy with the project.

## Artefacts

<a href="#"> Deployment</a>
<a href="#"> Testing</a>
<a href="#"> API Documentation</a>
<a href="#"> Deployment Guide</a>
<a href="#"> Server NGINX Guide</a>

# Deployment

## 26/08/2023: Initial Frontend Demo

- EC2 Instance spun up:
  - root email: [bmeachem@student.unimelb.edu.au](mailto:bmeachem@student.unimelb.edu.au), located at: [nish.expressionless.com.au](http://nish.expressionless.com.au)
- Nginx Configuration Deployed (No repository currently)
- Frontend Deployed, SHA: [7a95501759807420eb5a1b3a41ca995039052a4c](#)
- Backend not yet Deployed.
- Duration: 45 minutes (Incl. EC2 config, nginx config, frontend docker deployment)

### New Frontend Features:

- Introduced new Upload Window Modal with drag-and-drop capability
- reusable components for modals added
- file directory refactored for enhanced organization
- sample PDF items now displayable
- user feedback animations introduced
- New Item Upload Button implemented with click functionality.

### New Backend Features:

N/A

### New Devops Features:

- Started out on t2.micro EC2 instance
- Nginx config introduced
- Reverse Proxy on port 80 (Forwarded)
- Frontend working on Port 3001 through RP (Not Forwarded)

## 01/09/2023: EC2 Upgrade and Backend Start

- Duration: 1 hr and 20 minutes. (Incl initial attempt at using old EC2, upgrading to a fresh instance, re-setting up Docker, Nginx incl Config and Frontend + deploying minio and backend)
- EC2 Instance upgraded from t2.micro to t2.medium (1gb RAM to 4gb RAM)
- Nginx Configuration redeployed (No repository currently)
- Frontend Deployed, SHA: [7a95501759807420eb5a1b3a41ca995039052a4c](#)
- Backend Deployed, SHA: [759494580f4e6f32aa0a9237b3ac1a94226dc018](#)
- Minio Deployed, Docker Image: minio/minio

### New Frontend Features:

N/A

### New Backend Features:

- Upload/Download endpoints
- MinIO Deployment
- MinIO integration

- Keycloak Deployment

#### New Devops Features:

- Upgraded to t2.normal EC2 Instance

### 02/09/2023: (Hotfix) File Sizes

- Duration: 5 minutes
- Backend Deployed, SHA: [34679107ad8497feff085695d49bdd933a454219](#)

#### New Frontend Features:

N/A

#### New Backend Features:

- Allowed Larger File Size through Quarkus

#### New Devops Features:

- Allowed Larger File Size through Nginx

### 04/09/2023: Thumbnail Generation on Backend

- Duration: 5 minutes
- Backend Deployed, SHA: [143a3b5e1128db7649e695fd2862a56b21668b0a](#)

#### New Frontend Features:

N/A

#### New Backend Features:

- Generate Thumbnails on Upload of PDF
- Add List all PDF endpoint

#### New Devops Features:

N/A

### 9/10/2023: Backend Overhaul

- Duration: 5 minutes
- Backend Deployed, SHA: [781782b4ac483cd6bf25726647afdb6e1828a830](#)

#### Backend Features:

- New Database Models:
  - Story
  - StorySelection
  - Edition
  - Keyword
- New Endpoints

- /v2/story
  - POST /add/{edition-id} - Add a story
- /v2/edition]
  - POST /upload - Upload an Edition
  - GET /list - List existing Editions
- /v3/media
  - GET /public/fetch/pdf/{minio-id} - Fetch PDF
  - GET /public/fetch/tbn/{minio-id} - Fetch Thumbnail

#### **Frontend Features:**

- N/A

#### **New Devops Features:**

- CORS

### 17/10/2023: Updated Frontend Endpoint / Overhaul

#### **Backend Features:**

- Editions can now be published/drafted. Published Editions cannot be edited. Drafts maintain regular functionality
- Searching for Editions added
- Keywords Generate for editions
- Stories now store html along with delta and text
- Selections are now updated as a group of selections instead of individual selections
- Delete endpoints for stories and editions

#### **Frontend Features:**

- Basic Reader & Further Improvements To Routing [ For Reader Capability]
- Dynamic Background Colours
- Rectangles & Ordering On View Page Added
- Zoom Functionality & Scaling Rectangles with Zoom
- Updated Endpoints / New Endpoints in use
- Several Bug Fixes

#### **New Devops Features:**

- N/A

### 12/11/2023: Final Deployment

- Duration: 30 minutes
- Backend Deployed. SHA: [22f9d0dfd1910600c4022621c9a520d787f927b4](#)
- Frontend Deployed. SHA: [cd021408c0ed7631793a3a24e29fa199c34ca217](#)

**Backend Features:**

- Bug fixes
- Authentication

**Frontend Features:**

- A lot of Bug fixes
- Authentication
- Added Table of Contents for Reader
- Fixed Reader
- Added Toast messages for various events
- Added edit text button to go alongside regenerate text

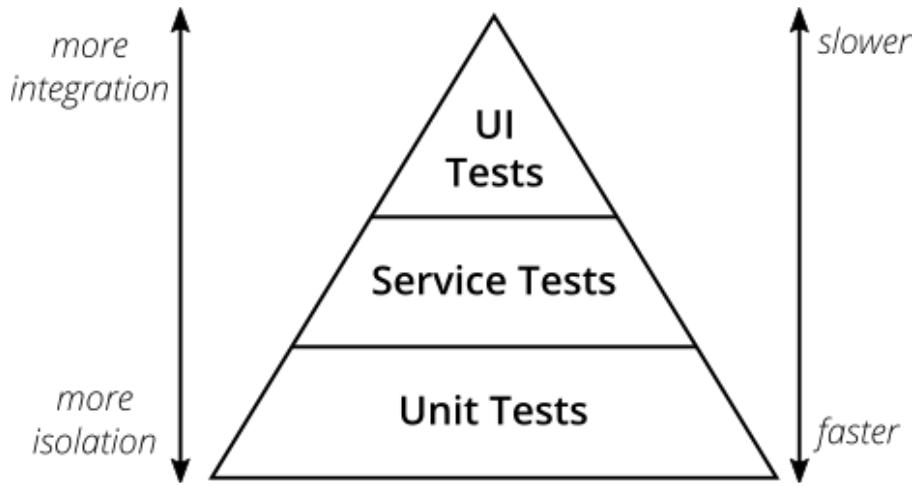
**Devops Features:**

- N/A

# Testing

## Testing Overview:

A practical approach to testing will be taken, with utilisation of Mike Cohn's **Test Pyramid**, designed as part of his book: **Succeeding with Agile**.



The overall premise is, when picking parts to test of a system, the more involved the test case is - the more the product gets tested overall, which is quite intuitive/straight forward. As this is a tradeoff that needs to be taken into account during testing, the team will carefully decide on a few UI/E2E tests, several service (integration) tests and a decent amalgam of unit tests.

### Some definitions:

- **UI/E2E (End to End) Tests:** Refers to (usually) manual work navigating through the product to test each part works, and in some cases attempting to break it as much as possible to test stability. This is usually taken care of by a dedicated Quality Assurance Team in a regular corporate environment, but we will still be able to perform these kind of tests
- **Service/Integration Tests:** Refers to (usually) automated testing within a particular service or core functionality. Think of this as **End to End testing of a single component**. In the frontend, typically this is done with **automated navigation of several smaller screens**, or ensuring that responses from endpoints are as anticipated. In the backend, typically this is done by **testing the endpoints and ensuring the responses are as expected, to a wider degree than the frontend**. This could also be **testing the connection to other services within the backend** (in our case the database or object storage). The reason that endpoint testing can take place on both the backend and frontend is that the backend can be capable of wider variety testing of endpoints, as those tests would be testing functionality. **On the frontend the endpoint tests** would simply be **testing a few results** to ensure that **reaching the endpoints actually work**. Since the endpoints are a backend construct, it is on the **backend to thoroughly test them**.
- **Unit Testing:** Refers to the **testing of singular methods** or a small group of methods. This is a lot less work to setup, as each test is much **smaller in scale compared to Integration or End to End tests**, and ensures that the **fundamentals of any given service are working as expected**. They are **purely isolated**, as they are **not intended for testing groups of functionality**. These usually quantify the largest portion of overall testing as it requires **a lot of unit tests to test the fundamentals of a service**. To add to this, by sufficiently unit testing individual pieces of functionality, tests that fail further up the chain can (usually) **rule out the breaking down of any single piece of functionality** as a reason for failing, since **the unit tests are responsible for single functionality testing**.

## End to End Testing:

At the end of each week, End to End testing of various user stories should occur to ensure that the functionality delivered by us meets the business requirements of the client in a (ideally) bug-free fashion. As this document progresses, user stories and test-paths will be added, and in later deployments these paths will be frequently referenced in them.

### User Stories to Test:

#### PDF Upload/Display Page (ADMINS)

- *Admins should be able to login to the page.*
  - Admins are redirected to a separate login page.
  - Admins enter their login details and log into the service.
  - Admins are redirected to the PDF display site.
- *Admins can view previously uploaded PDFs.*
  - PDFs are displayed as individual interactive thumbnails.
  - Admins can search for PDFs using keywords.
  - PDF thumbnails are resized as the Admin zooms in and out using zoom bar (TBD).
  - PDFs can be seen as uploaded (TBD), draft or published.
  - Clicking on PDF thumbnail should take you to the PDF Edit Page.
- *Admins can upload their own PDFs to the service.*
  - Alongside the PDFs is an interactive button for adding PDFs.
  - Clicking on the upload button should bring up a modal window.
  - Admins click the upload portion of the window and can upload files, viewing the uploaded file in the modal window.
  - Admins can click a button to upload the chosen file(s).
  - Admins can see the PDF has been added to their collection.

#### PDF View Page (USERS):

(NOTE)

- Users should be able to read PDF in **PDF** form.
  - Users should be able to read PDF in **TEXT** form.
  - Users should be able to switch between **PDF** and **TEXT** form on the fly.

## PDF Edit Page (ADMINS):

- Admins can switch between **VIEW** and **EDIT** mode.
  - In **VIEW** mode, Admins can view the page as if they were users.
    - See the *PDF View Page* user stories.
  - In **EDIT** mode, in **PDF** form, Admins can create/edit stories from the PDF that appear/will appear in the **TEXT** form.
  - In **EDIT** mode, in **TEXT** form, Admins can view and edit story text content/appearance.

## Frontend Testing:

- **Integration Testing:** The lack of asynchronous code (in the present) means we have not developed many integration tests. Currently there is a test that tests `react-router` navigates users to the correct page. However, these tests will be critical in future to determine if the endpoints are working correctly and will be developed as more Asynchronous is added. Testing these will be simple as we will utilise the same `react-testing-library` that is used for unit-tests to test these endpoints. The library provides helpers like `waitFor` to deal with Asynchronous actions, so that we are able to rigorously test fetching a pdf from an end-point or simulate uploading a file.
  - **Unit Testing:** The Frontend has utilised the `react-testing-library` to create unit tests for several custom components. These unit tests allow us to ensure they work in isolation correctly and to validate that each component performs the way it was designed. Not all components will receive unit testing as some will receive End to End testing since it is a more rigorous way to test certain UI-elements that are difficult to test through a singular method. Below are some unit test examples that are testing the state elements of the components:
    - NewUpload
    - PDFItem
    - UploadItem
    - UploadModal
  - **End To End Testing:** The Frontend has undergone some End-To-End testing for some of the Asynchronous Code added recently for the endpoints. This was primarily required as testing endpoints such as fetching text from a selection can be a tricky to complete using Integration or Unit Tests. Some procedures that were followed included testing various example selections and checking if the output was correctly. Other tests included checking on requests and response. sent through the Network Tab to see whether requests are being sent correctly and the correct respond is being sent. Below is an example of some end to end tests for the story endpoints. Other end to end tests was completed for the list endpoints to check whether correct PDF's were being found by the endpoint. This means of testing allowed us to find a bug regarding the React Hook `useEffect({...})` which was sending endless requests to the backend and flooding the server. It is critical that monitoring network activity be done when working the endpoints and hooks such as `useState(..)` and `useEffect({...})` do not overburden the server with requests. Most E2E testing involved monitoring network activity as it provided details on the requests sent and received.

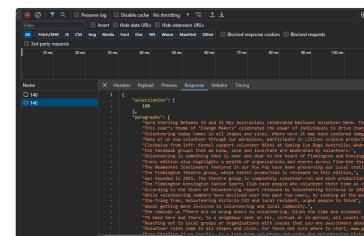
## Selection Example



## Response Example Console

```
1 {  
2     "selectionIds": [  
3         160  
4     ],  
5     "paragraphs": [  
6         "Sara Sterling Between 15 and 21 May Aust  
7         "This year's theme of 'Change Makers' cel  
8         "Volunteering today comes in all shapes a  
9         "Many of us now volunteer through our work  
10        "Clockwise from left: Kennel support volu  
11        "the Facebook groups that we know, love a  
12        "Volunteering is something that is near a  
13        "Every edition also highlights a wealth o  
14        "The Newmarket Stationeers featured in ou  
15        "The Flemington Theatre group, whose late  
16        "was founded in 2015. The theatre group i  
17        "The Flemington Kensington Junior Sports  
18        "According to the State of Volunteering r  
19        "While volunteering numbers have declined  
20        "Thu-Trang Tran, Volunteering Victoria CE  
21        "about getting more involved in volunteer  
22        "She reminds us "There are no wrong doors  
23        "A hand here and there, to a neighbour r  
24        "Reaching out to local groups or organisat  
25        "Volunteer roles come in all shapes and s  
26        "Sara Sterling lives locally, is a long-t  
27    ]  
28 }
```

## Monitoring Network Activity



```

src > tests > PDFItem.test.js > ...
1 import { render, screen, fireEvent } from '@testing-library/react';
2 import PDFItem from '../components/PDFUpload/PDFItem';
3 import { useNavigate } from 'react-router-dom';
4
5 jest.mock('react-router-dom', () => ({
6   useNavigate: jest.fn(),
7 }));
8
9 describe('<PDFItem />', () => {
10   it('renders image and text, and navigates on click', () => {
11     // Mock the useNavigate function
12     const mockNavigate = jest.fn();
13     useNavigate.mockReturnValue(mockNavigate);
14
15     render(<PDFItem />);
16
17     // Assert that the image and text are present
18     const imageElement = screen.getByAltText('Sample Episode');
19     const textElement = screen.getByText('Sample Episode');
20     expect(imageElement).toBeInTheDocument();
21     expect(textElement).toBeInTheDocument();
22
23     // Simulate a click on the component
24     fireEvent.click(screen.getByText('Sample Episode'));
25
26     // Assert that useNavigate was called with the expected path
27     expect(mockNavigate).toHaveBeenCalledWith('/pdf/12/tool/view');
28   });
29 });
30

```

Integration Test - checking the correct routing/redirection

```

1 import { render, screen, fireEvent, waitFor } from '@testing-library/react';
2 import NewUpload from '../components/PDFUpload/NewUpload';
3 import UploadItem from '../components/Modal/UploadModal/UploadItem';
4 describe('<NewUpload />', () => {
5   it('renders the upload button and opens modal when clicked', async () => {
6     render(<NewUpload />);
7
8     const uploadButton = screen.getByLabelText('Upload new PDF');
8     expect(uploadButton).toBeInTheDocument();
9
10    // Simulate a click on the button to open the modal
11    fireEvent.click(uploadButton);
12
13    const modalHeading = screen.getByText('Upload and attach files');
13    expect(modalHeading).toBeInTheDocument();
14
15    const closeButton = screen.getByLabelText('Close');
15    fireEvent.click(closeButton);
16
17    await waitFor(() => {
18      | expect(modalHeading).not.toBeInTheDocument();
19    });
20  });
21  it('uploads a file and checks its title in the modal', () => {
22
23    const file = new File(['hello'], 'hello.png', { type: 'image/png' })
23    render(<UploadItem fileData={file} />);
24
25    const fileNameElement = screen.getByText('hello.png');
25    expect(fileNameElement).toBeInTheDocument();
26
27  });
28 });
29
30
31
32
33 });
34

```

Unit Tests - checking whether a Modal is currently opening and a file upload is correctly displaying

## Backend Testing:

- **Integration Testing:** Working off a [Baeldung article on Unit and Integration Testing](#), it appears very possible to Integration test the backend's connection and usage of it's surrounding microservices (minio, keycloak and the database).
- **Unit Testing:** Unit tests are meant to be small and portable, and hence easy to write. In the backend, Quarkus is able to facilitate easy testing of endpoints as pictured below. For all other backend unit tests `@Brynn Meachem` has a custom framework that will work for repeat testing of the same single functionality which can be reused for different functionality testing (also below).

```

@QuarkusTest
public class GreetingResourceTest {

    @Test
    public void testHelloEndpoint() {
        given()
            .when().get("/hello")
        .then()
            .statusCode(200)
            .body(is("hello"));
    }

    @Test
    public void testGreetingEndpoint() {
        String uuid = UUID.randomUUID().toString();
        given()
            .pathParam("name", uuid)
        .when().get("/hello/greeting/{name}")
        .then()
            .statusCode(200)
            .body(is("hello " + uuid));
    }

}

```

2 Endpoint Unit Tests through Quarkus

```

@Test
public void removeFromArrayList() {
    new JUnitTester<String[], String>()
        .setTest((input) -> {
            String array = input.input[0];
            String[] entries = input.input[1] != null ? input.input[1].split(":") : null;
            return Expected(CommonUtils.removeFromArrayList(array, entries));
        })

        .createTest("Remove from array blank 1")
        .setInput(new String[] {
            "",
            "INPUT1:INPUT2"
        })
        .setExpected("[]")
        .build()

        .createTest("Remove from array blank 2")
        .setInput(new String[] {
            null,
            "INPUT1:INPUT2"
        })
        .setExpected("[]")
        .build()
}

```

Unit Testing a "remove from array" method that operates on json strings with 2 unit tests (the blank and null cases).

## Feature Testing:

[Feature Test Scenarios](#)

# Feature Test Scenarios

- Functional Requirements
- Non-Functional Requirements
- Business Requirements

## Functional Requirements

Feature	Description	Test Scenario	Accepted / Rejected
User View	The User view shall be publicly accessible without requiring user authentication and shall offer PDF search capabilities.	See PDF Viewer with Accessibility Features	
PDF Viewer with Accessibility Features	The User view's integrated PDF viewer shall offer accessibility features including searchable text.	<p>1. <b>Fetch Share Link</b></p> <ul style="list-style-type: none"> <li>◦ Navigate to the desired Edition</li> <li>◦ Click on the edit button</li> <li>◦ Click on the copy link button</li> <li>◦ Reader link should be copied to clipboard</li> </ul> <p>2. <b>Access Reader Site</b></p> <ul style="list-style-type: none"> <li>◦ Paste Copied URL to desired browser</li> <li>◦ Land on the User site without authentication</li> </ul> <p>3. <b>Check Table of Contents</b></p> <ul style="list-style-type: none"> <li>◦ Click on Table of Contents Icon</li> <li>◦ List of Stories should appear</li> <li>◦ Clicking on each story should navigate you to each story's text</li> </ul> <p>4. <b>Change Text Attributes</b></p> <ul style="list-style-type: none"> <li>◦ Click on Font Icon (Aa)</li> <li>◦ Drop down menu should appear</li> <li>◦ Clicking font increase should increase the font size</li> <li>◦ Clicking font decrease should decrease the font size</li> <li>◦ Clicking each color circle should change the color and text for ALL aspects of the viewer including PDF viewer, text, icons and table of contents.</li> <li>◦ Clicking on different font families should change the font for the TEXT only</li> </ul>	
Secondary Website Creation	The system shall create a secondary website accessible through a separate URL with distinct Admin and User views.	<p>Ensure these two URLs work</p> <p> <a href="#">React App</a>  <a href="https://keycloak.expressionless.com.au/">https://keycloak.expressionless.com.au/</a></p> <p>See PDF Viewer with Accessibility Features for the share link</p>	

<b>Search Functionality</b>	<p>The User view shall provide a keyword-based search feature for users to find relevant PDFs based on entered keywords.</p>	<ol style="list-style-type: none"> <li><b>1. Add Edition:</b> <ul style="list-style-type: none"> <li>◦ Open the application.</li> <li>◦ Navigate to the upload PDF modal</li> <li>◦ Use file selector to select file/ drag file into modal</li> <li>◦ Click on the "Upload" button.</li> <li>◦ Verify that the edition is successfully added.</li> </ul> </li> <li><b>2. Search for Edition using keywords in Edition Title (should be successful):</b> <ul style="list-style-type: none"> <li>◦ Navigate to the search functionality.</li> <li>◦ Enter keywords from the title of the previously added edition.</li> <li>◦ List of Editions should update as keyword is being typed.</li> <li>◦ Verify that the search results include the edition added in step 1.</li> </ul> </li> <li><b>3. Add Story to Edition, generating and saving text:</b> <ul style="list-style-type: none"> <li>◦ Click on the Edition added in step 1.</li> <li>◦ Click on edit icon.</li> <li>◦ Click on "New Story" button.</li> <li>◦ Click on PDF and create boxes around desired text.</li> <li>◦ Click on "Add Story".</li> <li>◦ Verify that the story is successfully added to the selected edition, and the text is saved.</li> </ul> </li> <li><b>4. Search for Edition using keywords in story (should be successful):</b> <ul style="list-style-type: none"> <li>◦ Navigate to the search functionality.</li> <li>◦ Enter keywords from the story added in step 3.</li> <li>◦ List of Editions should update as keyword is being typed.</li> <li>◦ Verify that the search results include the edition published in step 4.</li> </ul> </li> <li><b>5. Attempt to search for keywords in story (should be unsuccessful):</b> <ul style="list-style-type: none"> <li>◦ Navigate to the search functionality.</li> <li>◦ Enter keywords from the text of the story added in step 3.</li> <li>◦ Initiate the search.</li> <li>◦ Verify that the search results do not include the edition drafted in step 6.</li> </ul> </li> </ol>	
<b>User Authentication and Authorization</b>	<p>The system shall provide secure user authentication and authorization mechanisms with distinct roles: Admin and User.</p>	<p>See Admin View</p>	

<b>PDF Database</b>	<p>The system shall utilize a database to store PDF files along with associated metadata and extracted text for search.</p>	<ol style="list-style-type: none"> <li><b>1. Add Edition:</b> <ul style="list-style-type: none"> <li>◦ Open the application.</li> <li>◦ Navigate to the upload PDF modal</li> <li>◦ Use file selector to select file/ drag file into modal</li> <li>◦ Click on the "Upload" button.</li> <li>◦ Verify that the edition is successfully added:           <ul style="list-style-type: none"> <li>▪ PDF is in MinIO Storage.</li> <li>▪ Thumbnail is in MinIO Storage.</li> <li>▪ Both should have the same name.</li> </ul> </li> </ul> </li> <li><b>2. Add Story to Edition, generating and saving text:</b> <ul style="list-style-type: none"> <li>◦ Click on the Edition added in step 1.</li> <li>◦ Click on edit icon.</li> <li>◦ Click on “New Story” button.</li> <li>◦ Click on PDF and create boxes around desired text.</li> <li>◦ Click on “Add Story”.</li> <li>◦ Check that “Add Story“ redirects to the Quill Text Editor.</li> <li>◦ Verify that the story is successfully added to the selected edition, and the text is saved.           <ul style="list-style-type: none"> <li>▪ Delta and HTML files are created with the desired generated text in MinIO</li> <li>▪ New story row is added to database along with relevant information.</li> </ul> </li> </ul> </li> <li><b>3. Update Text:</b> <ul style="list-style-type: none"> <li>◦ Type new text into Quill Text Editor for the story added in step 2.</li> <li>◦ Click on the “Submit“ Button to save changes.</li> <li>◦ Check that the relevant files are updated as was done in step 2.</li> </ul> </li> <li><b>4. Update Selections:</b> <ul style="list-style-type: none"> <li>◦ Click on the Edition added in step 1.</li> <li>◦ Click on edit icon.</li> <li>◦ Click on the story added in step 2.</li> <li>◦ Click on PDF and create new boxes around desired text, delete old text boxes as well.</li> <li>◦ Click on Regenerate Text.</li> <li>◦ Verify that the story selections have been added to the desired story</li> <li>◦ Verify that the relevant files for the story are updated as was done in step 2.</li> </ul> </li> </ol>	
<b>Admin View</b>	<p>Admin users shall be able to log in using unique credentials to access the Admin view for PDF management.</p>	<ol style="list-style-type: none"> <li><b>1. Login:</b> <ul style="list-style-type: none"> <li>◦ Go to <a href="#">React App</a> <ul style="list-style-type: none"> <li>▪ New User:</li> </ul> </li> </ul> </li> </ol>	

		<ul style="list-style-type: none"> <li>• Should not be able to access website without logging in</li> <li>• See (2)</li> </ul> <p>▪ Admin User</p> <ul style="list-style-type: none"> <li>• Should not be able to access website if they have not logged in within 6 hours</li> <li>• Should be instantly taken to Admin View if they have logged in within 6 hours</li> </ul> <p><b>2. Create New User:</b></p> <ul style="list-style-type: none"> <li>◦ Follow Keycloak section in this video <a href="#">Flemington Kensington PDF Viewer Demo</a></li> </ul>	
<b>PDF Upload and Management</b>	Admin users shall be able to upload PDFs, edit metadata, and replace existing PDFs through the Admin view.	See PDF Database	
<b>Responsive UI Design</b> <b>Non-Functional Requirements</b>	The secondary website shall be designed to offer optimal viewing experiences on various devices and screen sizes.	See Search Functionality and User View	
Feature	Description	Test Scenario	Accepted / Rejected
<b>Error Handling</b>	The system shall provide informative error messages and user-friendly error handling mechanisms for user interactions.	<p><b>1. Add Edition:</b></p> <ul style="list-style-type: none"> <li>◦ Open the application.</li> <li>◦ Navigate to the upload PDF modal</li> <li>◦ Use file selector to select file/ drag file that is NOT a PDF into modal</li> <li>◦ Click on the "Upload" button.</li> <li>◦ Edition should not appear in the list of editions.</li> </ul>	
<b>Performance Optimization</b>	The secondary website's performance shall be optimized to ensure fast loading times and smooth user interactions.	<p>Refer to PDF Database, Search Functionality, PDF Upload and Manage.</p> <p>Each of these tests should be relatively quick to perform.</p>	

## Business Requirements

Feature	Description	Test Scenario	Accepted / Rejected
<b>Searchable PDFs</b>	The system shall extract text from PDFs and enable text search within the PDFs, enhancing search functionality.	<p>See the following Functional Requirement Tests:</p> <ul style="list-style-type: none"> <li>• Search Functionality</li> </ul>	
<b>Usability Enhancement</b>	The redesigned website shall improve user experience for individuals with disabilities and the elderly, increasing accessibility.	<p>See the following Functional Requirement Tests:</p> <ul style="list-style-type: none"> <li>• User View</li> </ul>	

<b>Keyword-Based PDF Locator</b>	The search functionality shall allow users to locate PDFs based on entered keywords, enhancing user experience.	See the following Functional Requirement Tests: <ul style="list-style-type: none"><li>• Search Functionality</li></ul>	
<b>User Roles and Privileges</b>	Admin users shall have elevated privileges for PDF management, while User users shall have read-only access.	See the following Functional Requirement Tests: <ul style="list-style-type: none"><li>• Admin View</li><li>• User View</li></ul>	

# API Documentation

## Description:

This document holds all of our endpoints, both in progress and completed. A full list of endpoints that are in use in current production can be found at: [OpenAPI UI \(Powered by Quarkus 2.16.10.Final\)](#)

## V1 API

### Public Resource

#### Version

Get the Version of the Backend Application

`GET /v1/api/public/version`

Responses

200 OK - version string in the response body

The screenshot shows the 'version' endpoint under the 'GET /v1/api/public/version' section. It includes sections for 'Parameters' (none), 'Responses' (200 OK), and 'Curl' (request and response examples). The response body contains '1.0.0-SNAPSHOT' and the response headers include 'connection: keep-alive', 'content-length: 14', 'content-type: text/plain; charset=UTF-8', 'date: Sat, 23 Sep 2023 16:11:20 GMT', and 'server: nginx/2.1.18.0 (Ubuntu)'.

## V2 API

### Minio Resource

#### Fetch PDF

Fetch a PDF from MinIO through the Backend

`GET /v2/media/public/fetch/pdf/{code}`

Path Parameters

`code` : The file name of the PDF to fetch

Responses

200 OK - if PDF exists with the PDF in the Response Body

GET /v2/media/public/fetch/pdf/{code}

**Parameters**

Name	Description
code <small>* required</small>	string (path)

**Responses**

**Curl**

```
curl -X 'GET' \
  'http://backend.expressionless.com.au/v2/media/public/fetch/pdf/202302WinterDigitalEDITED_1.PDF' \
  -H 'accept: */*'
```

**Request URL**

```
http://backend.expressionless.com.au/v2/media/public/fetch/pdf/202302WinterDigitalEDITED_1.PDF
```

**Server response**

Code	Details	Links
200	Response body <a href="#">Download file</a> Response headers <pre>connection: keep-alive content-disposition: attachment; filename="202302WinterDigitalEDITED_1.PDF" content-type: multipart/form-data date: Sat, 23 Sep 2023 14:20:07 GMT server: nginx/1.18.0 (Ubuntu) transfer-encoding: chunked</pre>	No links
200	OK	

404 NOT FOUND - if the PDF does not exist in the system

## Fetch Thumbnail of PDF

Fetch the thumbnail of a PDF from MinIO through the Backend

GET /v2/media/public/fetch/pdf/tbn/{code}

Path Parameters

code : The file name of the PDF of the thumbnail to fetch

Responses

200 OK - with the thumbnail in the response body if the PDF exists

404 NOT FOUND - if the PDF does not exist in the system

## List all PDFs

List all PDFs up to a (optional) limit in MinIO through the Backend

`GET /v2/media/public/list/pdf`

Query Parameters

`limit` : The maximum number of PDFs to return. (Default is max value of an Integer) OPTIONAL

Responses

200 OK - This will never not return an OK unless the Response is specified by Nginx or there is a connection issue

Sample Response:

```

1  {
2    "total": 3,
3    "size": 3,
4    "items": [
5      "202302WinterDigitalEDITED_1.PDF",
6      "IT_project_2.pdf",
7      "sample-pdf-with-images.pdf"
8    ]
9  }

```

**GET** /v2/media/public/list/pdf

**Parameters**

Name	Description
limit integer(\$int64) (query)	limit

**Responses**

**Curl**

```
curl -X 'GET' \
'http://backend.expressionless.com.au/v2/media/public/list/pdf' \
-H 'accept: */*
```

**Request URL**

```
http://backend.expressionless.com.au/v2/media/public/list/pdf
```

**Server response**

Code	Details
200	<b>Response body</b> <pre>{   "total": 3,   "limit": 0,   "size": 3,   "items": [     "20230923InterDigitalEDITED_1.PDF",     "IT_project_2.pdf",     "sample-pdf-with-images.pdf"   ] }</pre> <div style="display: flex; justify-content: space-end;"> <span>Copy</span> <span>Download</span> </div> <b>Response headers</b> <pre>connection: keep-alive content-length: 12 content-type: application/json date: Sat, 23 Sep 2023 14:18:42 GMT server: nginx/1.18.0 (Ubuntu)</pre>

**Responses**

Code	Description	Links
200	OK	No links

## Upload PDF

Upload a PDF to MinIO through the backend. Generates a Thumbnail to be fetched using the same file name as well

**GET** /v2/media/public/upload

### Responses

200 OK - If the upload was successful with the file name as a Plain String (no JSON structure) to be used for fetching that file

## Edition Resource

### List Editions

List all currently existing editions

**GET** /v2/edition/list

### Query Parameters

search-text : Text to be used for searching OPTIONAL

published-only : Whether or not the edition should show only published editions OPTIONAL

### Responses

200 OK - if editions that exist are successfully retrieved and returned

Sample Response:

```
1  {
2    "editions": [
```

```
3     {
4         "id": 6,
5         "fileName": "2023-Winter-Edition",
6         "author": "Milad Izra",
7         "minioId": "1a2cfaff-cbe6-49ae-b388-8e6f369b86e2",
8         "published": false
9     }
10    ]
11 }
```

## Upload Edition

Upload a file and create an edition associated with it

`POST /v2/edition/upload`

Responses

200 OK - if the PDF was uploaded and the edition was created

## Delete Edition

Delete an edition

`DELETE /v2/edition/{edition-id}`

Path Parameters

`edition-id` : id of edition to delete

Responses

204 NO CONTENT - Deletion was successful and there is no content left

## Publish Edition

Publish a draft edition

`PUT /v2/edition/publish/{edition-id}`

Path Parameters

`edition-id` : id of edition to publish

Responses

200 OK - If publishing was successful

## Draft Edition

Draft a published edition

`PUT /v2/edition/draft/{edition-id}`

Path Parameters

`edition-id` : id of edition to draft

Responses

200 OK - If drafting was successful

## Story Resource

### List Story

Lists stories related to an edition

`GET /v2/story/list/{edition-id}`

Path Parameters

`edition-id` : id of edition to add a new story to

Responses:

200 OK - if stories in an edition are successfully retrieved and returned

Sample Response

```
1 {
2   "stories": [
3     {
4       "id": 4,
5       "title": "Giga Mega Chaddington",
6       "author": "Jesus Christ",
7       "selections": [
8         {
9           "id": 3,
10          "x": 0,
11          "y": 0,
12          "width": 100,
13          "height": 100,
14          "pageNumber": 0,
15          "sequenceNumber": 1
16        }
17      ]
18    }
19  ]
20 }
```

### Add Story

Add story to an edition

`POST /v2/story/add/{edition-id}`

Path Parameters

`edition-id` : id of edition to add a new story to

Sample Request Body:

```

1 {
2   "title" : "Milad's Wacky Adventures",
3   "author" : "Milad Izra",
4   "selections" : [
5     {
6       "x" : 0,
7       "y" : 0,
8       "width" : 1000,
9       "height" : 1000,
10      "pageNumber" : 0,
11      "sequenceNumber" : 1
12    },
13    {
14      "x" : 10,
15      "y" : 10,
16      "width" : 1000,
17      "height" : 1000,
18      "pageNumber" : 0,
19      "sequenceNumber" : 2
20    }
21  ]
22 }

```

## Responses

200 OK - if story is successfully added

## Sample Response

```

1 {
2   "selectionIds": [
3     5,
4     6
5   ],
6   "paragraphs": [
7     "Sara Sterling Between 15 and 21 May Australians celebrated National Volunteer Week. The annual event recogn
8     "This year's theme of 'Change Makers' celebrated the power of individuals to drive change and ensure volunte
9     "Volunteering today comes in all shapes and sizes. Where once it may have conjured images of the local sport
10    "Many of us now volunteer through our workplace, participate in citizen science projects in our local parks,
11    "WINTER 2023",
12    "the Facebook groups that we know, love and love/hate are moderated by volunteers.",
13    "Volunteering is something that is near and dear to the heart of Flemington and Kensington residents. This p
14    "Every edition also highlights a wealth of organisations and events across Flem-Ken that are volunteer-drive
15    "The Newmarket Stationeers featured in our Vox Pop have been preserving our local station since 2010, and in
16    "The Flemington Theatre group, whose latest production is reviewed in this edition,",
17    "was founded in 2015. The theatre group is completely volunteer-run and each production is supported by a te
18    "The Flemington Kensington Junior Sports Club have people who volunteer their time as committee members, coa
19    "According to the State of Volunteering report released by Volunteering Victoria in 2020, approximately 2.3
20    "While volunteering numbers have declined over the past few years, by looking at the pages of this paper you
21    "Thu-Trang Tran, Volunteering Victoria CEO and local resident, urged people to think",
22    "about getting more involved in volunteering and local community.",
23    "She reminds us "There are no wrong doors to volunteering. Given the time and economic pressures on people,
24    ""A hand here and there, to a neighbour near or far, virtual or in-person, all counts to your wellbeing, you
25    "Reaching out to local groups or organisations with causes that you are passionate about is a great way to g
26    "Volunteer roles come in all shapes and sizes. For those not sure where to start, www.govolunteer.com.au is
27    "Sara Sterling lives locally, is a long-term volunteer and works for Volunteering Victoria."
28  ]
29 }

```

## Update Story Selections

Update selections of a story

```
PUT /v2/story/selections/update/{story-id}
```

Path Parameters

`story-id` : id of story that has selections to be updated

Sample Request Body:

```
1  {
2    "selections" : [
3      {
4        "x" : 0,
5        "y" : 0,
6        "width" : 1000,
7        "height" : 1000,
8        "pageNumber" : 0,
9        "sequenceNumber" : 1
10       },
11      {
12        "x" : 10,
13        "y" : 10,
14        "width" : 1000,
15        "height" : 1000,
16        "pageNumber" : 0,
17        "sequenceNumber" : 2
18      }
19    ]
20 }
```

Responses

200 OK - if selections are updated accordingly

Sample Response:

```
1  {
2    "selectionIds": [
3      5,
4      6
5    ],
6    "paragraphs": [
7      "Sara Sterling Between 15 and 21 May Australians celebrated National Volunteer Week. The annual event recogn
8      "This year's theme of 'Change Makers' celebrated the power of individuals to drive change and ensure volunte
9      "Volunteering today comes in all shapes and sizes. Where once it may have conjured images of the local sport
10     "Many of us now volunteer through our workplace, participate in citizen science projects in our local parks,
11     "WINTER 2023",
12     "the Facebook groups that we know, love and love/hate are moderated by volunteers.",
13     "Volunteering is something that is near and dear to the heart of Flemington and Kensington residents. This p
14     "Every edition also highlights a wealth of organisations and events across Flem-Ken that are volunteer-drive
15     "The Newmarket Stationeers featured in our Vox Pop have been preserving our local station since 2010, and in
16     "The Flemington Theatre group, whose latest production is reviewed in this edition",
17     "was founded in 2015. The theatre group is completely volunteer-run and each production is supported by a te
18     "The Flemington Kensington Junior Sports Club have people who volunteer their time as committee members, coa
19     "According to the State of Volunteering report released by Volunteering Victoria in 2020, approximately 2.3
```

```

20     "While volunteering numbers have declined over the past few years, by looking at the pages of this paper you
21     "Thu-Trang Tran, Volunteering Victoria CEO and local resident, urged people to think",
22     "about getting more involved in volunteering and local community.",
23     "She reminds us "There are no wrong doors to volunteering. Given the time and economic pressures on people,
24     ""A hand here and there, to a neighbour near or far, virtual or in-person, all counts to your wellbeing, you
25     "Reaching out to local groups or organisations with causes that you are passionate about is a great way to g
26     "Volunteer roles come in all shapes and sizes. For those not sure where to start, www.govolunteer.com.au is
27     "Sara Sterling lives locally, is a long-term volunteer and works for Volunteering Victoria."
28 ]
29 }
```

## Update Text Content

Update the relative fields for the text component of the PDF Viewer

**PUT** /v2/story/content/update/{story-id}

Path Parameters

`story-id` : id of story to update the text for

Sample Request Body:

```

1 {
2   "delta" : {
3     "ops" : [
4       {
5         "insert" : "Hello Sailor\n"
6       }
7     ]
8   },
9   "text" : "Hello Sailor",
10  "html" : "<p>Hello Sailor</p>"
11 }
```

Responses

200 OK - Text was successfully updated

## Get Delta

Retrieve delta for quill text editor

**GET** /v2/story/content/fetch/delta/{story-id}

Path Parameters:

`story-id` : id of story to fetch the delta of

Sample Response Body:

```

1 {
2   "delta" : {
3     "ops" : [
4       {
5         "insert" : "Hello Sailor\n"
6       }
7     ]
8   },
9 }
```

## Get Text

Retrieve text for quill text editor

```
GET /v2/story/content/fetch/text/{story-id}
```

Path Parameters:

`story-id` : id of story to fetch the raw text of

Sample Response Body:

```
1 {
2   "text" : "Hello Sailor"
3 }
```

## Get HTML

Retrieve html for quill text editor

```
GET /v2/story/content/fetch/html/{story-id}
```

Path Parameters:

`story-id` : id of story to fetch the html of

Sample Response Body:

```
1 {
2   "html" : "<p>Hello Sailor</p>"
3 }
```

## Delete Story

Delete story

```
DELETE /v2/story/{story-id}
```

Path Parameters

`story-id` : id of story to delete

Responses

204 NO CONTENT - if story is deleted successfully

## Minio Resource 2

### Fetch PDF

Fetch a PDF from MinIO through the Backend

```
GET /v3/media/public/fetch/pdf/{minio-id}
```

Path Parameters

`minio-id` : MinIO id of the PDF to fetch

Responses

200 OK - if PDF exists with the PDF in the Response Body

404 NOT FOUND - if the PDF does not exist in the system

## Fetch Thumbnail of PDF

Fetch the thumbnail of a PDF from MinIO through the Backend

```
GET /v3/media/public/fetch/pdf/tbn/{minio-id}
```

Path Parameters

`minio-id` : MinIO id of the thumbnail to fetch

Responses

200 OK - with the thumbnail in the response body if the PDF exists

404 NOT FOUND - if the PDF does not exist in the system

# Deployment Guide

This system can be deployed using docker for ease of use. Both the frontend and backend repositories have ready-to-go Dockerfiles and associated scripts for building the images. All that is needed is to build the image on the host machine or to build the image and push it to the host machine. Below is a step-by-step guide to execute this process.

It is recommended to use the latest version of Ubuntu Server to host the system.

For a full deployment to work with authentication, a running keycloak instance is also required. For keycloak to operate correctly, it must also be running through HTTPS (not HTTP). A step-by-step process for getting HTTPS to work on a deployed instance using NGINX will also be covered in this guide.

## Installing Required Software to run with Docker and First Steps

To build docker images, a version of java 17 and docker is required. This can be easily acquired using

1. Install docker for your system
  - a. [Install Docker Engine](#)
  - b. If using a headless linux environment (Ubuntu Server): [Install Docker Engine on Ubuntu](#)
2. Install docker-compose if on ubuntu: [Install the Compose plugin](#)

Now that Docker is installed, go ahead and sort out [Server NGINX Guide](#) before continuing.

## Building the backend docker image

The backend docker image can be built on any system, provided that the architecture

1. Install sdkman: [Home - SDKMAN! the Software Development Kit Manager](#)
  - a. linux: `curl -s "https://get.sdkman.io" | bash`
  - b. Other systems: [Installation - SDKMAN! the Software Development Kit Manager](#)
2. Install the recommended version of java for this project:
  - a.  `sdkman.sdk install java 17.0.8-graalce`
3. Clone the backend repository found at [GitHub - FKNEditor/backend](#)
  - a. run `./build-docker.sh`

## Building the frontend docker image

The Frontend Docker image must be built on a system that has the same architecture as the system that will be hosting it. (Mac M1 will not build images for linux in this regard.)

### Steps

The first two steps can be handled with [GitHub - nvm-sh/nvm: Node Version Manager - POSIX-compliant bash script to manage multiple active node.js versions](#) if on ubuntu, which will allow easy version management the same way that [sdkman.io](#) does. If not building on an ubuntu system and not using ubuntu server, installing node and npm through [Node.js](#) should be fine as well.

1. Install node v18.17.0 & npm v9.6.7 or higher
2. Clone the frontend repository found at [GitHub - FKNEditor/frontend](#)
3. Create a `.env` file with the following:

```
1 REACT_APP_BACKEND_URL=<BACKEND_URL>
2 REACT_APP_KEYCLOAK_REALM=<KEYCLOAK REALM>
3 REACT_APP_KEYCLOAK_CLIENT_ID=<CLIENT ID>
4 REACT_APP_KEYCLOAK_URL=<KEYCLOAK URL>/auth
```

Example:

```
1 REACT_APP_BACKEND_URL="https://backend.expressionless.com.au"
2 REACT_APP_KEYCLOAK_REALM="itproject"
3 REACT_APP_KEYCLOAK_CLIENT_ID="frontend"
4 REACT_APP_KEYCLOAK_URL="https://keycloak.expressionless.com.au/auth"
```

4. run `./build-docker.sh`. It is important that the `.env` file containing the necessary variables is in place before the docker image is built, otherwise the frontend will not be able to connect to the backend API or to Keycloak.

## Running the system using Docker Compose

The system can be run quite easily using a docker-compose.yml. To run the system the following services are required to be run:

- MinIO
- Keycloak & a database for keycloak to connect to
  - See: [Keycloak Realm Configuration Guide](#) to configure the Keycloak instance once it is running
- A PostgreSQL database for the backend system to connect to
- the backend api

All of these services have docker images associated with them, so they can be run in containers alongside the API service.

Sample docker-compose.yml:

```
1 version: '3.7'
2 services:
3   keycloak-db:
4     container_name: keycloak-db
5     image: mysql:8.0.27
6     ports:
7       - 3319:3306
8     restart: unless-stopped
9     environment:
10       # The user, password and database that Keycloak
11       # is going to create and use
12       MYSQL_USER: keycloak_user
13       MYSQL_PASSWORD: <PASSWORD TO KEYCLOAK DATABASE>
14
15       MYSQL_DATABASE: keycloak_db
16       MYSQL_ROOT_PASSWORD: <PASSWORD TO KEYCLOAK DATABASE>
17     volumes:
18       - keycloak-mysql-vol:/var/lib/mysql
19
20   project-keycloak:
21     container_name: project-keycloak
22     image: jboss/keycloak:16.1.1
23     ports:
24       - 8180:8080
25     restart: unless-stopped
26     environment:
27       # This will be the login for the admin console of keycloak
28       # This is used for creating new users
29       - KEYCLOAK_USER=admin
30       - KEYCLOAK_PASSWORD=<CONSOLE PASSWORD>
31
32       - DB_VENDOR=mysql
33       - DB_ADDR=keycloak-db
34       - DB_PORT=3306
```

```

35      - DB_DATABASE=keycloak_db
36
37      - DB_USER=keycloak_user
38      - DB_PASSWORD=<PASSWORD TO KEYCLOAK DATABASE>
39
40      - PROXY_ADDRESS_FORWARDING=true
41      - KEYCLOAK_LOGLEVEL=debug
42      - JAVA_OPTS_APPEND="-Djava.awt.headless=true"
43      - PREPEND_JAVA_OPTS=-Dkeycloak.profile=preview -Dkeycloak.profile.feature.token_exchange=enabled -Dkeycloak.profile.feature.claim-exchange=enabled
44 depends_on:
45     - keycloak-db
46
47 postgresdb:
48     container_name: postgresdb
49     image: postgres:14-alpine
50     ports:
51         - 6969:5432
52     volumes:
53         - postgres_data:/var/lib/postgresql/data
54     environment:
55         # Make sure to set this and the project-backend DATABASE_PASSWORD to be the same
56         - POSTGRES_PASSWORD=<DATABASE PASSWORD>
57         - POSTGRES_USER=admin
58         - POSTGRES_DB=db
59
60 minio:
61     container_name: minio
62     image: minio/minio
63     ports:
64         - "9000:9000"
65         - "9001:9001"
66     volumes:
67         - minio_storage:/data
68     environment:
69         MINIO_ROOT_USER: admin
70         # Make sure these are the same as IT_PROJ_MINIO_PASS
71         MINIO_ROOT_PASSWORD: <PASSWORD TO PDF STORAGE>
72     command: server --console-address ":9001" /data
73
74 project-backend:
75     image: nishitproject/backend:backend
76     container_name: backend
77     restart: always
78     environment:
79         DATABASE_URL: postgresdb
80         DATABASE_USER: admin
81         DATABASE_PASSWORD: <DATABASE PASSWORD>
82         CORS_URLS: <SET THIS TO BE THE FRONTEND URL, WHERE THE FRONTEND SERVICE WILL BE MAKING REQUESTS FROM!>
83         # e.g https://keycloak.expressionless.com.au
84         UNI_KEYCLOAK_URL: <Set this to be the root URL of the keycloak instance through https>
85         ITPROJ_MINIO_URL: http://minio:9000
86         ITPROJ_MINIO_USER: admin
87         ITPROJ_MINIO_PASS: <PASSWORD TO PDF STORAGE>
88
89         PY_SCRIPT_DIR: /deployments/pyscripts/
90         NLTK_DATA: /usr/local/lib/nltk_data
91     ports:
92         - 8181:8080

```

```
93  depends_on:
94    - postgresdb
95    - minio
96  frontend:
97    stdin_open: true
98    container_name: frontend
99    image: nishitproject/backend:frontend-dev
100   volumes:
101     - './:/app'
102     - '/app/node_modules'
103   ports:
104     - 3001:80
105     - 8080:8080
106   environment:
107     - CHOKIDAR_USEPOLLING=true
108
109 volumes:
110   keycloak-mysql-vol:
111   postgres_data:
112   minio_storage: {}
```

# Server NGINX Guide

NGINX is a fantastic tool for server management. In the way it is used for this project, it provides minimal exposure to the internet. Below is a set of instructions on using NGINX with Ubuntu Server.

## Installing NGINX on Ubuntu Server

1. [Install | NGINX](#) citing this article, to install on ubuntu server simply do:
  - a. `sudo apt update`
  - b. `sudo apt install nginx`
2. Once this is done there should be new files located at `/etc/nginx`
3. New configuration (such as the one below) should be placed in `/etc/nginx/sites-available`. See below "Sample HTTP Configuration" for a configuration to place here (to replace `default`). Note that specific entries that are inside `<>` should be replaced with legitimate values (E.g `KEYCLOAK_URL` ).
4. The status of nginx server can be seen with this command: `sudo systemctl status nginx`

```
Unknown command verb nginx.  
ubuntu@ip-172-31-7-36:/etc/nginx/sites-available$ sudo systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy server  
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)  
  Active: active (running) since Sat 2023-10-07 02:47:56 UTC; 1 month 5 days ago  
    Docs: man:nginx(8)  
   Main PID: 517 (nginx)  
     Tasks: 3 (limit: 4667)  
       Memory: 23.1M  
          CPU: 11min 23.131s  
        CGroup: /system.slice/nginx.service  
            └─ 517 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"  
              ├─ 15366 "nginx: worker process"  
              ├─ 15367 "nginx: worker process"  
Oct 07 02:47:55 ip-172-31-7-36 systemd[1]: Starting A high performance web server and a reverse proxy server...  
Oct 07 02:47:56 ip-172-31-7-36 systemd[1]: Started A high performance web server and a reverse proxy server.  
ubuntu@ip-172-31-7-36:/etc/nginx/sites-available$
```

An active nginx deployment

5. [How To Install Nginx on Ubuntu 20.04 | DigitalOcean](#) referencing this guide, it is a good idea to check the firewall status on the server.
  - a. First, `sudo ufw status`. If this shows Status: inactive then `sudo ufw enable` will enable it
  - b. `sudo ufw app list` should show a list of available applications to allow through the firewall. E.g:  
Available applications:  
Nginx Full  
Nginx HTTP  
Nginx HTTPS  
OpenSSH
  - c. `sudo ufw allow 'Nginx Full'` will add Nginx to the list of allowed applications through the ubuntu firewall.
  - d. This can be verified by again checking the status: `sudo ufw status`

```
Rule added (v6)  
ubuntu@ip-172-31-7-36:/etc/nginx/sites-available$ sudo ufw status  
Status: active  
  
To                         Action      From  
--                         --         --  
Nginx Full                  ALLOW      Anywhere  
OpenSSH                     ALLOW      Anywhere  
Nginx Full (v6)              ALLOW      Anywhere (v6)  
OpenSSH (v6)                 ALLOW      Anywhere (v6)
```

An active Ubuntu firewall, with nginx and ssh allowed through the firewall

6. Now that Nginx is configured, `sudo nginx -t` will check that the current configuration is valid. If there are issues then this will let you know if there are any problems with it.

```
ubuntu@ip-172-31-7-36:/etc/nginx/sites-available$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

7. Finally, `sudo nginx -s reload` will bring the changes into effect

## Enabling HTTPS

 How To Secure Nginx with Let's Encrypt on Ubuntu 22.04 | DigitalOcean

Ensure snap is updated

1. `sudo snap install core; sudo snap refresh core`

Install Certbot

2. `sudo snap install --classic certbot`

Ensure Certbot can be run from the command line with `certbot`

3. `sudo ln -s /snap/bin/certbot /usr/bin/certbot`

Get certificates for the relevant domains

4. `sudo certbot --nginx`. This will open up a prompt to ask which domains should have certificates associated with them. These domains should come from the server configuration that was set up in the above steps. The domains should point to a legitimate domain, such as [google.com](https://google.com). This way the certificates can be linked to a trusted certificate authority. Press enter to select all of the domains (recommended)

After this, you should be prompted to enter an email address and agree to the terms of service. After doing so, you should see a message telling you the process was successful and where your certificates are stored.

```
Last login: Sun Nov 12 08:10:53 2023 from 180.150.36.177
ubuntu@ip-172-31-7-36:~$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
-----
1: expressionless.com.au
2: backend.expressionless.com.au
3: frontend-dev.expressionless.com.au
4: keycloak.expressionless.com.au
5: minio.expressionless.com.au
6: nish.expressionless.com.au
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```

## Sample HTTP Configuration

The below configuration is setup to use HTTP (on port 80). This will need Certificates to work with HTTPS. Certificates can be created by following the above steps

```
1 server {
2     root /var/www/html;
3
4     # Add index.php to the list if you are using PHP
5     index index.php index.html index.htm index.nginx-debian.html;
6
7     server_name <INSERT FRONTEND URL HERE>;
8     location / {
9         proxy_pass http://localhost:3001;
10        proxy_redirect off;
11    }
12
13    listen 80;
14 }
15
16 # This server Block is here only if external access to the pdf storage console is
17 # needed. This is not recommended in production
18 server {
19     server_name <INSERT MINIO URL>;
20     location / {
21         proxy_pass http://localhost:9001;
22         proxy_redirect off;
```

```

23     proxy_set_header Host $http_host;
24     proxy_set_header X-Real-IP $remote_addr;
25     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
26     proxy_set_header X-Forwarded-Proto $scheme;
27     proxy_set_header X-NginX-Proxy true;
28
29     # This is necessary to pass the correct IP to be hashed
30     real_ip_header X-Real-IP;
31
32     proxy_connect_timeout 300;
33
34     # To support websocket
35     proxy_http_version 1.1;
36     proxy_set_header Upgrade $http_upgrade;
37     proxy_set_header Connection "upgrade";
38
39     chunked_transfer_encoding off;
40 }
41
42 listen 80;
43 }
44
45 server {
46     # Keycloak Url, e.g: keycloak.expressionless.com.au
47     # note no http or https!!! That is handled by the server block itself
48     server_name <KEYCLOAK URL>;
49
50     proxy_set_header Host $host;
51     proxy_set_header X-Real-IP $remote_addr;
52     proxy_set_header X-Forwarded-Proto $scheme;
53     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
54     location / {
55         proxy_pass http://localhost:8180;
56         proxy_redirect off;
57     }
58
59     listen 80;
60 }
61
62 server {
63     client_max_body_size 50M;
64     access_log logs/nginx.log;
65     proxy_hide_header X-Frame-Options;
66
67     # This should be set to the same url as the frontend is expecting the backend
68     # to be located at
69     # e.g backend.expressionless.com.au
70     server_name <BACKEND URL>;
71     location / {
72         proxy_pass http://localhost:8181;
73         proxy_redirect off;
74     }
75
76     listen 80;
77 }
```

# Keycloak Realm Configuration Guide

## Getting Started

Getting started in Keycloak can be quite daunting, so this guide is here to help you through it. Once logged in, using the credentials defined with the `KEYCLOAK_USER` and `KEYCLOAK_PASSWORD` variables you will be greeted with this page.

The screenshot shows the Keycloak interface for the 'Master' realm. On the left, a sidebar lists 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', and 'Authentication'. Below that is a 'Manage' section with 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The main content area is titled 'Master' and has tabs for 'General', 'Login', 'Keys', 'Email', 'Themes', 'Localization', 'Cache', 'Tokens', 'Client Registration', and 'Client Policies'. The 'General' tab is selected. It contains fields for 'Name' (set to 'master'), 'Display name' (set to 'Keycloak'), 'HTML Display name' (containing the Keycloak logo), 'Frontend URL' (empty), 'Enabled' (set to 'ON'), 'User-Managed Access' (set to 'OFF'), 'User Profile Enabled' (set to 'OFF'), and 'Endpoints' (listing 'OpenID Endpoint Configuration' and 'SAML 2.0 Identity Provider Metadata'). At the bottom are 'Save' and 'Cancel' buttons.

Keycloak Page after login

It won't have `Itproject` in the list of realms yet, so go ahead and create it by mousing over `Master` and clicking Add realm. You will be brought to a form asking for a realm name. We named ours Test for this guide.

The screenshot shows the Keycloak interface for the newly created 'Test' realm. The sidebar and tabs are identical to the 'Master' realm. The 'General' tab is selected, showing a form with 'Name' set to 'test', 'Display name' empty, 'HTML Display name' empty, 'Frontend URL' empty, 'Enabled' set to 'ON', 'User-Managed Access' set to 'OFF', 'User Profile Enabled' set to 'OFF', and 'Endpoints' listing 'OpenID Endpoint Configuration' and 'SAML 2.0 Identity Provider Metadata'. At the bottom are 'Save' and 'Cancel' buttons.

Post-creation of a realm

Great! You have your first realm. Now you need to setup a client for users to connect to your realm through. Go ahead and select clients from the left and click create in the table.

The screenshot shows the 'Clients' table in the 'Test' realm. The sidebar shows 'Configure', 'Realm Settings', and 'Clients'. The table has columns for 'Client ID' (with entries 'account' and 'account-console'), 'Enabled' (both set to 'True'), 'Base URL' (with values 'https://keycloak.expressionless.com.au/auth/realms/test/account/' and 'https://keycloak.expressionless.com.au/auth/realms/test/account/'), and 'Actions' (with 'Edit', 'Export', and 'Delete' buttons). A 'Create' button is located at the top right of the table.

To successfully create a new client, you should specify the url that they are connecting from. For us this is <https://nish.expressionless.com.au>. Make sure also to note down the client id as it will be needed to configure the frontend service later.

## Add Client

Import

Client ID \*

Client Protocol

Root URL

## Adding a client

Now that you have a client. It is time to configure it. Note the Root URL can be changed at any time in the main page of the settings. Below that is **Valid Redirect URIs**. To make it possible to see the frontend after logging in, you should add a **Valid Redirect URI** that is the same as your **Root URL** except it has **\*** appended to it (e.g [https://nish.expressionless.com.au/\\*](https://nish.expressionless.com.au/*)). You should do the same for **Web Origins** as well.

Clients > frontend

Frontend

Settings Keys Roles Client Scopes Mappers Scope Revocation Sessions Offline Access Installation Permissions

Client ID: frontend

Name:

Description:

Enabled: ON

Always Display In Console: OFF

Consent Required: OFF

Login Theme:

Client Protocol: openid-connect

Access Type: public

Standard Flow Enabled: ON

Implicit Flow Enabled: OFF

Direct Access Grants Enabled: ON

OAuth 2.0 Device Authorization Grant Enabled: OFF

Front Channel Logout: OFF

Root URL: <https://nish.expressionless.com.au/>

\* Valid Redirect URIs: [https://nish.expressionless.com.au/\\*](https://nish.expressionless.com.au/*)

## Setting a redirect URI

Root URL

\* Valid Redirect URIs

Base URL

Admin URL

Logo URL

Policy URL

Terms of service URL

Web Origins

## Setting a valid Web Origin

Great! That's it. You have configured Keycloak for your new deployment of the Flemington Kensington News PDF Editor. Here are some other neat things you can do with the service.

## Increasing the time before log out

Currently the system will invalidate your session after 5 minutes. If you want to increase this, you can go to the Realm Settings and in the Tokens tab you can increase the `Access Token Lifespan`. A word of caution, the longer this is, the more insecure it can be as it provides more time for bad guys to get your token!

The screenshot shows the Keycloak admin interface for a realm named "Test". The left sidebar has sections for Configure (Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication) and Manage (Groups, Users, Sessions, Events, Import, Export). The main content area is titled "Test" and shows the "Tokens" tab selected. Under "Default Signature Algorithm", RS256 is chosen. The "Revoke Refresh Token" switch is off. The "SSO Session Idle" dropdown is set to 30 Minutes. The "SSO Session Max" dropdown is set to 10 Hours. The "SSO Session Idle Remember Me" dropdown is set to 0 Minutes. The "SSO Session Max Remember Me" dropdown is set to 0 Minutes. The "Offline Session Idle" dropdown is set to 30 Days. The "Offline Session Max Limited" switch is off. The "Client Session Idle" dropdown is set to 0 Minutes. The "Client Session Max" dropdown is set to 0 Minutes. The "Access Token Lifespan" dropdown is set to 5 Minutes, with an orange arrow pointing to it. Below it, "Access Token Lifespan For Implicit" is set to 15 Minutes.

Increasing the time on sessions

## Adding New Users

Adding new users is quite simple. Go to the Users section of your new realm and click add user. You will be greeted with a new form, and the only thing you need to fill in is the `Username`. This is the thing they'll use to log in, and the system doesn't use any of the other things so they are very optional.

The screenshot shows the Keycloak admin interface for a realm named "Test". The left sidebar has sections for Configure (Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication) and Manage (Groups, Users). The "Users" section is selected, indicated by an orange arrow. The main content area shows the "Users" tab selected and the "Add user" form. The form fields include: ID (empty), Created At (empty), Username (set to "someusernamehere"), Email (empty), First Name (empty), Last Name (empty), User Enabled (ON), Email Verified (OFF), Groups (dropdown menu showing "Select existing group..." with "No group selected"), Required User Actions (dropdown menu showing "Select an action..."), and Save and Cancel buttons at the bottom.

Now that you have created the new user, it is time to set a password so that they can login! Go ahead and click on the Credentials tab in the new user you just created. Here you can specify a new password and optionally make it temporary. Setting it to temporary means they will need to reset their password on first login.

The screenshot shows the Keycloak administration interface. The top navigation bar has 'Users' and 'someusernamehere'. Below it, tabs include 'Details', 'Attributes', 'Credentials' (which is active), 'Role Mappings', 'Groups', 'Consents', and 'Sessions'. Under 'Manage Credentials', there's a table header with columns: Position, Type, User Label, Data, and Actions. Below this is a 'Set Password' section with fields for 'Password' and 'Password Confirmation', both empty. A 'Temporary' checkbox is checked and labeled 'ON'. At the bottom is a 'Set Password' button.

That's it. Congratulations, you now have a fully setup Keycloak instance. On to the next thing!

[Deployment Guide](#)