**Project Reference Document**

**Contents**

1. [Application Idea](#Application_Idea)
2. [Requirements](#Requirements)
3. [Architecture](#Architecture)
4. [Technology Stack](#Technology_Stack)
5. [Roadmap](#Roadmap)
6. [Application Evaluation](#Application_Evaluation)
7. [Log Book](#Log_Book)
8. **Application Idea**

**Words Wide Web**: This web application enables users to identify the trend of word/s from news on a specific given date (past or present) or range of dates across the World Wide Web. Furthermore, it may provide to the users, storing of the findings on a server along with statistical analysis capabilities.

1. **Requirements**

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-001 - Web App** |
| **Goal** | The Web App shall enable the users to identify the trend of word/s across the World Wide Web. |
| **Event** | - |
| **Precondition** | - |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | HIGH |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-002 - Responsive Web Design** |
| **Goal** | Dynamic appearance of the Web content based on media screen sizes for desktops and tablets shall be considered. |
| **Event** | - |
| **Precondition** | - |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-003 - Web App Response Time** |
| **Goal** | The response time of the Web App when a user interacts with it shall be without noticeable delays. |
| **Event** | - |
| **Precondition** | - |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | LOW |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-004 – REST WEB API** |
| **Goal** | A REST Web API service provided by a backend server shall enable registered users to store/read search results. |
| **Event** | - |
| **Precondition** | The REST Web API shall provide data in JSON format. |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-005 – REST WEB API Limited Operations** |
| **Goal** | The REST Web API service shall provide just the CRD (CREATE | READ | DELETE) operations to the user. |
| **Event** | - |
| **Precondition** | - |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-006 – SQLite Database** |
| **Goal** | The search results shall be stored in a SQLite Database (on the same server as the WEB API) when there is a need from the user side. The storage shall follow the below JSON structure:  [  {  **record\_id**: *string,*  **search\_text**: *string,*  **user\_id**: *string,*  **results**: [  {  **published\_date**: *string*  **source**: {id: *string*, name: *string*}  **description**: *string*  **url**: *string*  },  {  **published\_date**: *string*  **source**: {id: *string*, name: *string*}  **description**: *string*  **url**: *string*  },  {  …..  }  ]  },  {  **record\_id**: *string,*  **search\_text**: *string,*  **user\_id**: *string,*  **results**: […]  }  ]  The above JSON structure shall be a four column table in the database:  **record\_id**  **search\_text**  **user\_id**  **results**  A separate table shall hold the total available requests (UPDATE operation):  **requests\_limit** |
| **Event** | - |
| **Precondition** | - |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-007 – News API** |
| **Goal** | The search engine of the Web App is based on the News API: <https://newsapi.org/s/google-news-api> that shall be consumed on the client side. |
| **Event** | - |
| **Precondition** | The News API shall provide data in JSON format. |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-008 – News API Limitations** |
| **Goal** | The News API: <https://newsapi.org/s/google-news-api> shall have a limit of 100 requests per days for all registered users all together. |
| **Event** | - |
| **Precondition** | - |
| **Standard Flow** | - |
| **Alternative Flow** | - |
| **Post-condition Success** | - |
| **Post-condition Error** | - |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-009 – REST WEB API Error Handling** |
| **Goal** | On no access to the server due to e.g. bad request, unauthorized etc. an error should be shown |
| **Event** | Try to POST the Search Results |
| **Precondition** | REST WEB API Server is running |
| **Standard Flow** | CREATED 201 |
| **Alternative Flow** | NOT OK 200 AND NOT CREATED 201 e.g. Error 4xx, 5xx |
| **Post-condition Success** | Info popup “Info! The search results have been saved successfully” |
| **Post-condition Error** | Error popup “Error! No Server Access. The results cannot be saved” |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |
| **ID - Name** | **FAFE-010 – News API Error Handling** |
| **Goal** | On no access to the server due to e.g. limit exceeded, traffic overload etc. an error should be shown |
| **Event** | Try to GET the Search Results |
| **Precondition** | News API Server is running |
| **Standard Flow** | OK 200 |
| **Alternative Flow** | NOT OK 200 e.g. Error 4xx, 5xx |
| **Post-condition Success** | Search results are received successfully |
| **Post-condition Error** | Error message proportional to News API response object: {status, code, message} |
| **Classification** | **Non-Functional, CAN** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-011 – Context Selection Drop Down List** |
| **Goal** | A user shall be able to select from a drop down list what context area must be included for the search attempt. The List shall consist of the following entries:  **Everything**  **Top headlines**  **Business**  **Entertainment**  **General**  **Health**  **Science**  **Sports**  **Technology** |
| **Event** | On drop down list caret click, the list shall fade in. |
| **Precondition** | Context selection list has already a pre-selected or a default value. |
| **Standard Flow** | On selecting of an item, the list shall fade out. |
| **Alternative Flow** | On not selecting an item, the list shall automatically fade out by keeping the previous selection active. |
| **Post-condition Success** | Displays the selected item. |
| **Post-condition Error** | - |
| **Classification** | **Functional, MUST** |
| **Effort** | LOW |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-012 – From To Date Input Fields** |
| **Goal** | There shall be two Date input fields to enable the user to select the date search range (From and To). |
| **Event** | On Date input click, a Calendar shall fade in |
| **Precondition** | - |
| **Standard Flow** | The **From** and **To** inputs shall have a valid or zero positive difference (To shall be equal or greater than From) and the max difference shall be limited to 1 month. |
| **Alternative Flow** | On nothing selected or in case the difference is negative, the smallest date is set for both inputs |
| **Post-condition Success** | Both input fields display a valid date |
| **Post-condition Error** | Automatically set the corresponding input to today’s value |
| **Classification** | **Functional, MUST** |
| **Effort** | LOW |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-013 – Stored Results Drop Down List** |
| **Goal** | A registered user shall be able to select from a drop-down list stored results where the list entries shall follow this naming model:  **FromDate1-ToDate2-Context Name-Increment**  e.g. 11022023-15022023-Science-01  The entry name shall be stored in database entry **id** (see **FAFE-006**). |
| **Event** | On drop down list caret click, the list shall fade in. |
| **Precondition** | Context selection list has already a pre-selected or a default value. |
| **Standard Flow** | On selecting of an item, the list shall fade out. |
| **Alternative Flow** | On not selecting an item, the list shall automatically fade out by keeping the previous selection active. |
| **Post-condition Success** | Displays the selected item |
| **Post-condition Error** | - |
| **Classification** | **Functional, MUST** |
| **Effort** | LOW |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-014 – Save Results Button** |
| **Goal** | A registered user shall be able to store the results in the SQLite Database on the server by clicking on Save Results Button. |
| **Event** | On clicking the Save Results |
| **Precondition** | Results must be available |
| **Standard Flow** | Results are saved to the SQLite database |
| **Alternative Flow** | On no results available, the Save Results Button shall be disabled |
| **Post-condition Success** | Info popup “Success!” |
| **Post-condition Error** | Error popup “Error! Results cannot be saved” |
| **Classification** | **Functional, MUST** |
| **Effort** | LOW |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-015 – Web Search Bar** |
| **Goal** | The Web Search Bar shall consist of an input field that enables the registered user to enter a text and then to press the Search Button. |
| **Event** | On Search Button click |
| **Precondition** | Input field must have a text and the user must have logged in |
| **Standard Flow** | Entered Text is posted successfully |
| **Alternative Flow** | On empty input field or not logged in user, the Search Button shall be disabled |
| **Post-condition Success** | Results are listed as a table like structure |
| **Post-condition Error** | Error popup is shown |
| **Classification** | **Functional, MUST** |
| **Effort** | LOW |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-016 – Displaying the Search Results** |
| **Goal** | The search results shall be displayed as a table like the one structure/styling shared below:   * **Table Head**:Id, Results Count: xxx * **Collapsible Content**: The data cell below the Results Count shall display a collapsible content grouped by the source name |
| **Event** | On Search Button click |
| **Precondition** | Search Results must be available |
| **Standard Flow** | Search Results are displayed as a table like structure |
| **Alternative Flow** | On no Results, no table must be shown |
| **Post-condition Success** | Search Results are displayed as a table like structure |
| **Post-condition Error** | Info popup “No results found” or Error popup “No Server Access” |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-017 – Stored Results Chart** |
| **Goal** | A registered user shall be able to display the trend of a selected stored Result in a graphical manner.  Type of Chart: **Vertical Bar**  X: **Date**  Y: **Occurrence** (Search Text) |
| **Event** | On selecting an item as described by the **FAFE-013** |
| **Precondition** | Stored results must be available |
| **Standard Flow** | A stored Result is selected |
| **Alternative Flow** | On nothing selected, no Chart is shown on screen |
| **Post-condition Success** | The Chart of the selected stored Result is shown on screen |
| **Post-condition Error** | Error “No Server Access” |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-018 – Login** |
| **Goal** | A login form shall allow users to create an account using the following input type/description:  **Text: Email**  **Password: Password**  **Submit: Login**  **Link: Register => Dependency: FAFE-019** |
| **Event** | On Login button press, the user shall be logged in to the server |
| **Precondition** | The user must have been registered first and the login form fields shall be valid (button is disabled for invalid inputs) |
| **Standard Flow** | The user is logged in successfully to the server |
| **Alternative Flow** | On not registered user, an error shall be shown on screen |
| **Post-condition Success** | A “Login successful” message shall appear on screen |
| **Post-condition Error** | An Error “Email or Password is invalid” shall appear on screen |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-019 – Register** |
| **Goal** | A login form shall allow users to create an account using the following input type/description:  **Text: Email**  **Password: Password**  **Submit: Register** |
| **Event** | On Register button press, the user shall be registered to the server |
| **Precondition** | The registration form input fields shall be valid (button is disabled for invalid inputs) |
| **Standard Flow** | The user is registered successfully to the server |
| **Alternative Flow** | On invalid inputs, the button shall be disabled |
| **Post-condition Success** | A “Registration successful” message shall appear on screen |
| **Post-condition Error** | An Error “Email or Password is invalid” shall appear on screen |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

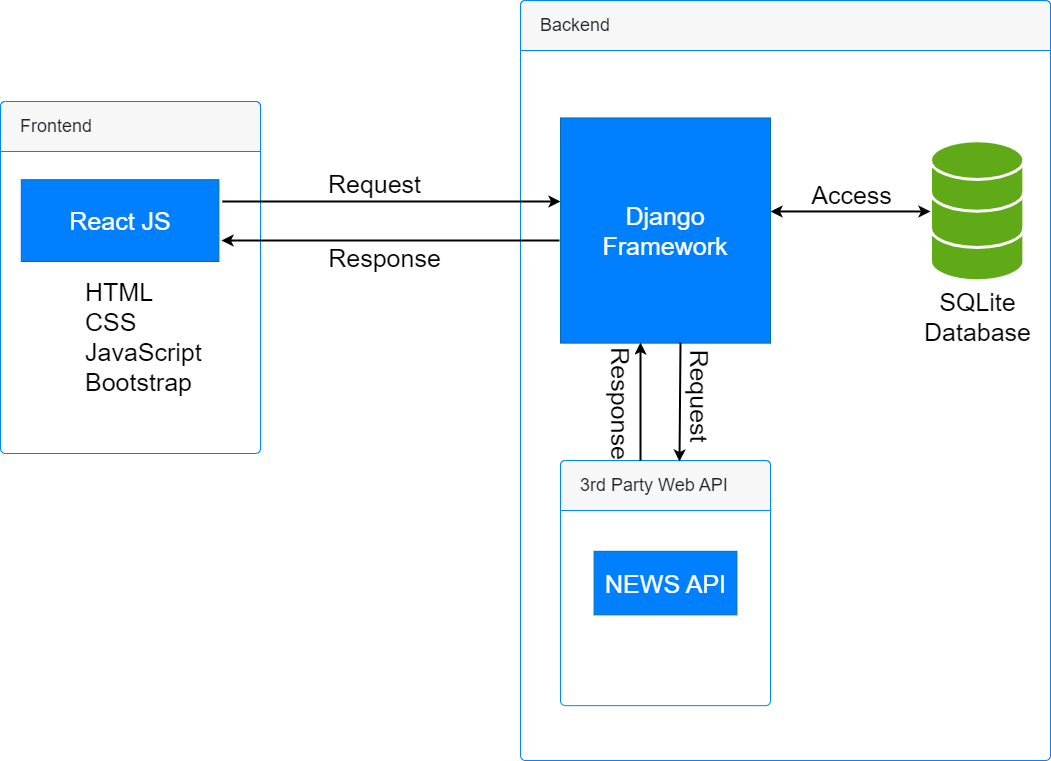
|  |  |
| --- | --- |
| **ID - Name** | **FAFE-20 – Home Page** |
| **Goal** | The website shall have a Navigation bar with links to **Home**, Word Search, History and Login Page. The **Home** Page shall have a structure according to the **Home** Page mockup: |
| **Event** | On Home Link press |
| **Precondition** | Frontend App must be running |
| **Standard Flow** | The user is redirected to the Home Page |
| **Alternative Flow** | Redirect to Error Page |
| **Post-condition Success** | Home Page is shown on screen |
| **Post-condition Error** | Error Page and message “Page Not Found” |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-21 – Word Search Page** |
| **Goal** | The website shall have a Navigation bar with links to Home, **Word Search**, History and Login Page. The **Word Search** Page shall have a structure according to the **Word Search** Page mockup: |
| **Event** | On Word Search Link press |
| **Precondition** | Frontend App must be running |
| **Standard Flow** | The user is redirected to the Word Search Page |
| **Alternative Flow** | Redirect to Error Page |
| **Post-condition Success** | Word Search Page is shown on screen |
| **Post-condition Error** | Error Page and message “Page Not Found” |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-22 – History Page** |
| **Goal** | The website shall have a Navigation bar with links to Home, Word Search, **History** and Login Page. The **History** Page shall have a structure according to the **History** Page mockup: |
| **Event** | On History Link press |
| **Precondition** | Frontend App must be running |
| **Standard Flow** | The user is redirected to the History Page |
| **Alternative Flow** | Redirect to Error Page |
| **Post-condition Success** | History Page is shown on screen |
| **Post-condition Error** | Error Page and message “Page Not Found” |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

|  |  |
| --- | --- |
| **ID - Name** | **FAFE-23 – Login/Register Page** |
| **Goal** | The website shall have a Navigation bar with links to Home, Word Search, History and Login/Register Page. The **Login/Register** Page shall have a structure according to the **Login/Register** Page mockup: |
| **Event** | On Login/Register Link press |
| **Precondition** | Frontend App must be running |
| **Standard Flow** | The user is redirected to the Login/Register Page |
| **Alternative Flow** | Redirect to Error Page |
| **Post-condition Success** | Login/Register Page is shown on screen |
| **Post-condition Error** | Error Page and message “Page Not Found” |
| **Classification** | **Functional, MUST** |
| **Effort** | MEDIUM |
| **Status** | **Open** |

1. **Architecture (**[**Back to Contents**](#Contents)**)**

****

1. **Technology Stack (**[**Back to Contents**](#Contents)**)**

Frontend: **React JS**, **React** **Bootstrap**

Database: **SQLite**

Backend: **Django Rest Framework, 3rd Party API (News API)**

**Client Side Packages (Dependencies)**

|  |  |
| --- | --- |
| **"@reduxjs/toolkit": "^1.9.5",**  **"@testing-library/jest-dom": "^5.16.5",**  **"@testing-library/react": "^13.4.0",**  **"@testing-library/user-event": "^13.5.0",**  **"autoprefixer": "10.4.5",**  **"axios": "^1.4.0",**  **"bootstrap": "^4.6.0",**  **"chart.js": "^4.3.0",**  **"formik": "^2.2.9",**  **"react": "^18.2.0",** | **"react-bootstrap": "^2.7.4",**  **"react-chartjs-2": "^5.2.0",**  **"react-dom": "^18.2.0",**  **"react-icons": "^4.8.0",**  **"react-redux": "^8.0.5",**  **"react-router-bootstrap": "^0.26.2",**  **"react-router-dom": "^6.10.0",**  **"react-scripts": "5.0.1",**  **"web-vitals": "^2.1.4",**  **"yup": "^1.1.1"** |

**Server Side Packages (Dependencies)**

|  |  |
| --- | --- |
| **appdata==2.1.2**  **asgiref==3.6.0**  **attrs==22.2.0**  **autopep8==2.0.1**  **black==23.1.0**  **certifi==2022.12.7**  **charset-normalizer==3.1.0**  **click==8.0.3**  **colorama==0.4.6**  **colored==1.4.3**  **Django==4.1.7**  **django-cors-headers==3.14.0**  **django-environ==0.10.0**  **django-filter==22.1**  **django-json-field==0.5.7**  **django-jsonfield==1.4.1**  **django-queryable-properties==1.8.4**  **djangorestframework==3.14.0**  **djangorestframework-simplejwt==5.2.2**  **dnspython==2.3.0**  **drf-spectacular==0.26.1**  **idna==3.4**  **inflection==0.5.1** | **jsonschema==4.17.3**  **mypy-extensions==1.0.0**  **newsapi-python==0.2.7**  **packaging==23.0**  **pathspec==0.11.1**  **platformdirs==3.1.1**  **pycodestyle==2.10.0**  **PyJWT==2.6.0**  **pyrsistent==0.19.3**  **python-dateutil==2.8.2**  **pytz==2022.7.1**  **PyYAML==6.0**  **requests==2.28.2**  **six==1.16.0**  **sqlparse==0.2.4**  **to-requirements.txt==1.3.0**  **tzdata==2022.7**  **uritemplate==4.1.1**  **urllib3==1.26.15** |

1. **Roadmap (**[**Back to Contents**](#Contents)**)**

|  |  |
| --- | --- |
|  |  |

1. **Application Evaluation (**[**Back to Contents**](#Contents)**)**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Implementation**  **Satisfaction** | **Comments** |
| **FAFE-001** | **█ █ █ █ █** | The App provides and stores the search results along with a Graph representation of the trend of the target text in the News |
| **FAFE-002** | **█ █ █ █ █** | Satisfactory responsiveness under all media |
| **FAFE-003** | **█ █ █ █ █** | Minor app’s delays appear during the UI Update. Cause: Implementation of the Redux Store |
| **FAFE-004** | **█ █ █ █ █** | Implementation through the Django Rest Framework |
| **FAFE-005** | **█ █ █ █ █** | DELETE from CRD Operations is not implemented |
| **FAFE-006** | **█ █ █ █ █** | A better structure has been implemented. Table: record\_id, title, content (JSON Field) grouped by publishedAt, owner\_id |
| **FAFE-007** | **█ █ █ █ █** | A 3rd party Web API integrated into the Django Web API and consumed on the client side (ReactJS) |
| **FAFE-008** | **█ █ █ █ █** | The free features are limited to 100 requests per day. No UI Counter has been implemented. |
| **FAFE-009** | **█ █ █ █ █** | Satisfactory Error Handling has been implemented |
| **FAFE-010** | **█ █ █ █ █** | Satisfactory Error Handling has been implemented |
| **FAFE-011** | **█ █ █ █ █** | Implemented |
| **FAFE-012** | **█ █ █ █ █** | Max difference is limited to 28 days. |
| **FAFE-013** | **█ █ █ █ █** | The entry name is slightly different as follow e.g. 0009:2023-05-20 to 2023-06-17:Science: quantum => record\_id:fromDate to toDate: text |
| **FAFE-014** | **█ █ █ █ █** | Implemented |
| **FAFE-015** | **█ █ █ █ █** | Implemented |
| **FAFE-016** | **█ █ █ █ █** | A better data representation has been implemented. Collapsible Table: Source-Title having the results grouped by date |
| **FAFE-017** | **█ █ █ █ █** | Implemented |
| **FAFE-018** | **█ █ █ █ █** | Username, Password and Login Button |
| **FAFE-019** | **█ █ █ █ █** | A better implementation has been implemented: Username, E-mail, Password, Password Confirmation and Register Button |
| **FAFE-020** | **█ █ █ █ █** | Implemented |
| **FAFE-021** | **█ █ █ █ █** | A slightly different UI has been implemented |
| **FAFE-022** | **█ █ █ █ █** | A slightly different UI has been implemented based on a collapsible table and fixed Graph |
| **FAFE-023** | **█ █ █ █ █** | A slightly different UI has been implemented |

1. **Log Book (**[**Back to Contents**](#Contents)**)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Time** | **Task** | **Comment** |
| **08.02.2023** | **-** | **Create the Project Reference Document Structure (PRD)** |  |
| **09.02.2023** | **-** | **Brainstorming: Full-Stack Web App Project** |  |
| **09.02.2023** | **-** | **Define the Non-Functional Requirements** |  |
| **13.02.2023** | **-** | **Define the Non-Functional Requirements** |  |
| **14.02.2023** | **-** | **Define the Functional Requirements** |  |
| **15.02.2023** | **-** | **Define the Full-Stack Web App Architecture** |  |
| **15.02.2023** | **-** | **Create the Application Evaluation Table** |  |
| **15.02.2023** | **-** | **Create the Log Book Table** |  |
| **15.02.2023** | **-** | **Create an empty README.md (Installation Instructions)** |  |
| **04.03.2023** | **-** | **Requirements Improvement based on PLA Feedback** |  |
| **05.03.2023** | **-** | **Architecture Improvement based on PLA Feedback** |  |
| **07.03.2023** | **-** | **Creation of Mockup Web Pages** |  |
| **08.03.2023** | **-** | **Requirements FAFE-020, FAFE-021, FAFE-022, FAFE-023 added** |  |
| **17.03.2023** | **-** | **React Boilerplate** |  |
| **18.03.2023** | **-** | **Django with REST API Framework Installed Boilerplate** |  |
| **23.03.2023** | **-** | **Experiment with Django REST API Framework** |  |
| **24.03.2023** | **-** | **How To Instructions Notes Creation for the Django REST API** |  |
| **25.03.2023** | **-** | **Commit Django REST API Instructions to GitHub** |  |
| **25.03.2023** | **-** | **Basic Django API with GET POST Operations and Access to SQLite** |  |
| **08.04.2023** | **-** | **Understanding/Testing JWT Authentication** |  |
| **09.04.2023** | **-** | **Understanding JWT Authentication within a Django Web API** |  |
| **10.04.2023** | **-** | **How To Instructions Notes Creation for the Django REST API with JWT Authentication** |  |
| **11.04.2023** | **-** | **JWT Authentication Integration into Project** |  |
| **22.04.2023** | **-** | **Understanding/Testing React Router** |  |
| **23.04.2023** | **-** | **Understanding/Testing React Redux** |  |
| **24.04.2023** | **-** | **Integrating React Router** |  |
| **25.04.2023** | **-** | **Integrating React Redux** |  |
| **26.04.2023** | **-** | **Integrating Axios JWT Interceptor** |  |
| **27.04.2023** | **-** | **Integrating React Bootstrap and Implementing Menu Navigation** |  |
| **29.04.2023** | **-** | **Implementing the Login Form in React and Access to Django API** |  |
| **11.05.2023** | **-** | **Completion of the Login and Register Page** |  |
| **12.05.2023** | **-** | **Implementing the Word Search Page** |  |
| **13.05.2023** | **-** | **Project Dependencies Documentation** |  |
| **14.05.2023** | **-** | **README.md: Installation Instructions Completed** |  |
| **15.05.2023** | **-** | **Refactoring and Block Comments** |  |
| **16.05.2023** | **-** | **Integrating Top Headlines Route into Django API** |  |
| **17.05.2023** | **-** | **Implementing data restructuring (grouping) on received API data** |  |
| **18.05.2023** | **-** | **Testing different ways of how to display the Search results** |  |
| **20.05.2023** | **-** | **Complete Implementation of displaying table like collapsible Search results** |  |
| **04.06.2023** | **-** | **Introduction of the content JSON Field in Record Model** |  |
| **05.06.2023** | **-** | **Successful Test with Thunder Client in Creation/Read of Records having JSON Content** |  |
| **06.06.2023** | **-** | **Implementing the Home Page** |  |
| **15.06.2023** | **-** | **Implementing a common View for the endpoints /api/records/ and /api/records/{id}** |  |
| **17.06.2023** | **-** | **Implementing a Save button to save the search results into backend db.sqlite3 database** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **17.06.2023** | **-** | **Implementing the History Page (Display selected Record and its Graph)** |  |
| **18.06.2023** | **-** | **Final Tests and Project Completion** |  |