

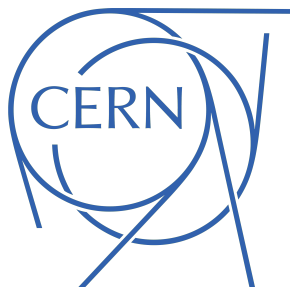
# CAiMIRA WP4 - Frontend of first NeutrinoReview Prototype

## Summer student report

Mario Negovetić<sup>1</sup>

Supervisor: Elias Sandner<sup>2</sup>

August, 2024



### **Abstract**

The summer student project described in this report addresses the frontend development of a web application, called NeutrinoReview. It aims to guide users through the systematic review process and utilizes a Large Language Model (LLM) to automate the title and abstract screening phase. The project is conducted in the context of the CERN and WHO joint venture CAiMIRA/ARIA and is a part of Work Package 4 (WP4) which addresses research and development of systematic review automation. The outcome of this project is a first frontend prototype, covering all required features as well as a simulated backend, returning static values to be able to demonstrate the frontend functionality.

---

<sup>1</sup>Graz University of Technology, Graz, Austria

<sup>2</sup>CERN IT-Department

# 1 Introduction

Systematic review (SR) is the research methodology that provides the most possible evidence to address a research question[1]. This is achieved by combining the findings of a maximum number of existing studies. However, this high evidence comes at the costs of a time-consuming process. Systematic reviews can be applied in any research domain, though it is mostly used in medical research. The main reason for this is that in the medical domain, high evidence is of especially high relevance. The first systematic review was done in 1753 by the medical researcher James Lind [2]. To reduce the workload of systematic reviews, several research projects tried to apply artificial intelligence (AI) and natural language processing (NLP) to streamline and automate specific tasks.

Besides others, researchers from The European Organization for Nuclear Research (CERN) and World Health Organization (WHO) are contributing to the field of systematic review automation by developing an review management tool that will reduce human involvement in the entire process. However, automation of the complete process does not seem to be feasible at the current time. Nevertheless, some steps could be automated to a stage where only human observation is required, while in others, support systems could speed up the human workflow.

In the scope of the NeutrinoReview project, this summer student project addressed the development of the first prototypes frontend. This first prototype is limited to the automation of the title and abstract screening. The whole application is created in the JavaScript framework React<sup>1</sup> with the help of open-source libraries. Additionally, to support the application logic, a backend simulation was developed to imitate a database and send data using specific endpoints. There is a total of 15 endpoints consisting of simple functions that return static data to the frontend. The web application described in this report consists of front- and back-end parts. However, while the backend functionality is currently under development, this report details the efforts for developing the frontend of the first prototype.

This report has six chapters. This chapter describes the main ideas behind SR and this project. The second chapter, Systematic Reviews, explains the main steps in SR. The third chapter, CaiMIRA, describes the main project for whose context this application was developed. Finally, in the Project structure chapter, the folder structure of the project is described, while in the Implementation chapter, the development of the application frontend is described. The chapter Conclusion emphasizes the main findings from developing this application and discusses possible future work.

---

<sup>1</sup><https://react.dev/>

## 2 Systematic Reviews

A systematic review is a process to evaluate existing research to answer a research question with sufficient evidence[3]. It minimizes bias and provides a comprehensive overview of the empirical evidence which are connected to a research question. Moreover, the main goal is to test a certain one or series of hypotheses [4]. However, without an automation process, SRs are time-consuming and can take several months to 2 years [5].

Despite being time-consuming, it is a process with a lot of benefits. Considering the high workload, systematic reviews might not be the perfect methodology in every situation. However, the main reason to use it is that looking at just one isolated study can lead to wrong answers, so it is more useful to look at more studies and results on certain topics. This approach is particularly pronounced in today's time as a large number of scientific publications are available and need to be properly filtered [4]. Another good reason is that SRs are useful when there is an abundance of research on a given topic but a lack of conclusive results. Often, the existing studies cover a wide range of aspects within a subject, making it difficult for researchers to determine where to focus their efforts. By systematically collecting, evaluating, and summarizing the available literature, SRs help to clarify which questions have already been addressed and, more importantly, which remain unresolved. This process allows researchers to identify and concentrate on the open research questions that still need exploration, guiding future studies toward filling these gaps in knowledge rather than merely replicating existing work [6].

### 2.1 Main stages

Systematic Reviews involve multiple stages as illustrated in 1 and described subsequently. The process can be divided into the main stages which are marked with blue color, while the grey fields refer to preparation and follow-up actions. As a preparation, it is important to do pre-research to specify a research question and figure out if SR is the appropriate research methodology. The next step is to develop a project proposal. It offers a detailed outline of the research objective, hypothesis, search terms, and criteria to include and exclude studies. A frequently applied approach to specify the eligibility criteria is to follow the PICO framework [7]. It consists of four questions: the population of interest, the intervention, the comparison of interventions, and the relevant outcomes. It is important to carefully choose inclusion criteria because all screening decisions are made solely based on these predefined criteria. The search strategy is defined based on the previous step where the user specifies search terms. The recommendation is to use at least two databases in which a search strategy will be applied based on the combination of text terms or code terms. After the database searching, the user can start with the title and abstract screening (TiAb). This step should involve at least two reviewers to minimize human bias. The screening processes purpose is to filter all studies that do not fulfill the predefined eligibility criteria. Following this step, papers considered as relevant based on the title and abstract are subject of an additional screening step, but based on the full text of those studies. Once all irrelevant papers have been filtered out, the process continues with the extraction of relevant information from all included studies. After these main steps, the user can write a review and submit a scientific paper.



Figure 1: The steps in systematic review. The blue steps are main while the grey are additional

### 3 CAiMIRA

The CAiMIRA (CERN Airborne Model for Risk Assessment) project was started in the spring of 2020 to simulate the virus concentration in closed environments. The main focus is on the SARS-CoV-2 virus because the initial goal was to understand the risk of long-range airborne spread of the virus. The CAiMIRA model is created based on publications in the fields of airborne transmission of infectious diseases, virology, epidemiology, and aerosol science [8].

#### 3.1 Conceptual design of NeutrinoReview

The CAiMIRA model relies on more than 100 parameters and with the current work-intensive systematic review process it is not feasible to retrieve all of them through SRs. Therefore, it was decided to expand the scope of the ARIA/CAiMIRA project with WP4, addressing efforts to streamline the process while remaining the quality. In this context, a novel review management tool called NeutrinoReview is under development. It is intended to automate certain steps and provide guidance for those steps where automation is not feasible yet.

The conceptual architecture of NeutrinoReview is illustrated in 2 and subsequently detailed. As an initial step, the user will be guided through an input form, asking for all required information. Next, the user has the opportunity to use the integrated feature for creating the search string in an iterative and ML supported manner. Next, the selected databases are queried automatically, and the retrieved studies are deduplicated. After deduplication, the remaining records will be automatically screened using a foundational LLM. Only records where the LLM cannot make a clear exclusion decision will require manual screening by the users. Since automating the Title and Abstract (TiAb) screening is the most critical aspect of this project, it has been prioritized and is the sole focus of the first prototype. The final NeutrinoReview tool will then also provide an automated full-text retrieval and a semi-automated full-text screening feature. In the end, data extraction will be done using the support of AI.

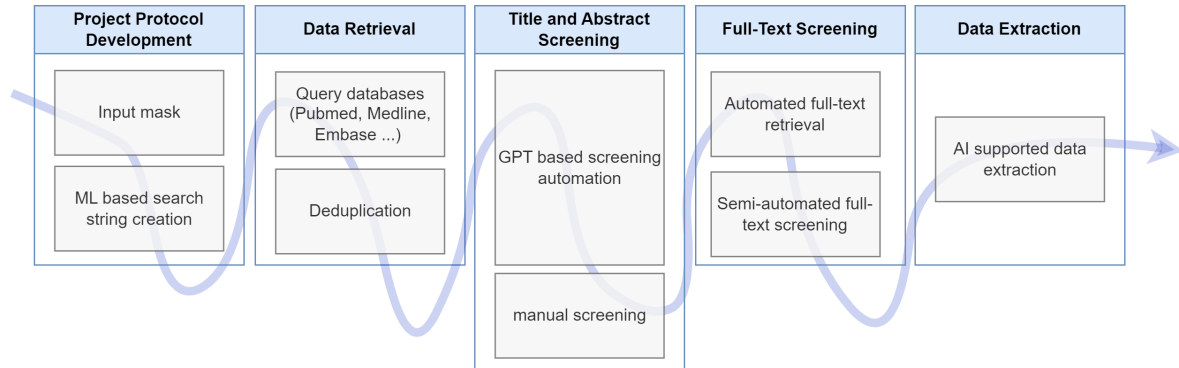


Figure 2: The architecture of new systematic review NeutrinoReview

## 4 Project structure

While it is the goal of WP4 of the CAiMIRA project to address the whole SR process from creating the project proposal up to extracting data from relevant papers, the development is divided into several stages. It starts with a prototype limited to the title and abstract screening phase because this was identified as the highest priority. The scope of this summer student project is limited to the development of the UI and together with the results of ongoing research about the most promising automation approaches, it will serve as a basis for the associated backend implementation.

### 4.1 Backend

The backend is designed using the Flask<sup>2</sup> micro-framework. It consists of an environment folder that contains all the necessary scripts and libraries so that the Python code can be interpreted. To accompany the imported scripts, an app.py file containing static datasets to imitate a database and 15 separate methods was created. Each method represents one endpoint that serves to retrieve, filter, and send data back to the frontend. The logic created within the methods is simple and serves to demonstrate the frontend as if it would be a real use case.

All API endpoints are listed below:

- GET /users/{userID}/reviews
- DELETE /users/{userID}/reviews/{reviewID}/delete
- POST /users/{userID}/reviews
- POST /users/{userID}/reviews/{reviewID}/autoScreen
- GET /users/{userID}/reviews/{reviewID}/autoScreen/progress
- GET /users/{userID}/reviews/{reviewID}/eligibilityCriteria
- GET /users/{userID}/reviews/{reviewID}/manualScreening/nextPaper
- POST /users/{userID}/reviews/{reviewID}/manualScreening/{citationID}/decision
- GET /users/{userID}/reviews/{reviewID}/resultsManual
- GET /users/{userID}/reviews/{reviewID}/results
- GET /users/{userID}/reviews/{reviewID}/results/csv
- GET /users/{userID}/reviews/{reviewID}/autoScreen/progress/action
- POST /users/{userID}/reviews/{reviewID}/edit
- GET /users/{userID}/reviews/{reviewID}/manualScreening/nextArticle
- GET /users/{userID}/reviews/{reviewID}/manualScreening/previousArticle

---

<sup>2</sup><https://flask.palletsprojects.com/en/3.0.x/>

## 4.2 Frontend

The frontend is implemented in React<sup>3</sup> using Vite framework, following the usual and recommended structure where pages are split on small components. The main file is App.jsx, containing the components and pages included in the router. The remaining files are structured in the following directories: actions, API, assets, components, context, pages and utils. The schema of frontend structure can be seen on Figure 3.

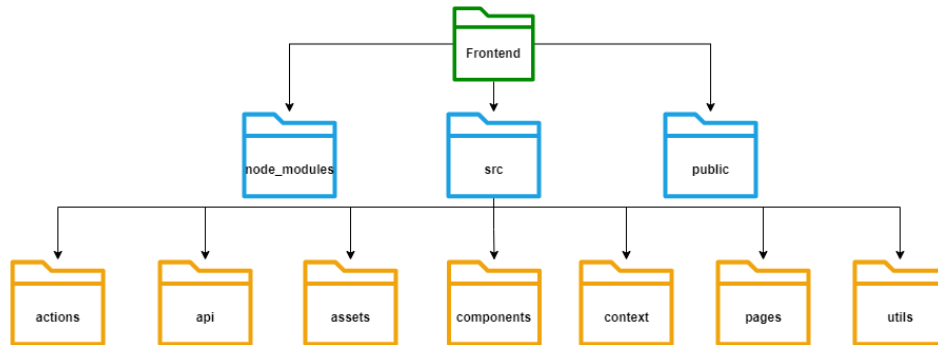


Figure 3: The scheme of frontend structure

Actions and API folder contain all endpoints that are called asynchronously using the Axios package. A separate function has been created for each endpoint since the response received from the backend is sent back to certain components.

The assets directory contains static content such as images. The utils directory contains all the constants, such as the names of the steps, the local URI, and the name of all open-source packages. Pages and component folders are mutually dependent. Components consist of smaller parts such as the footer, header, Prisma diagram, eligibility criteria table, and others. These components represent individual parts that possess individual logic.

The components are combined in page files that represent different views of the web-application. By integrating components, reusability is given which is also the main purpose in React projects. The context folder is also important, containing variables that are available in all steps in the Stepper component. All components have access to all these variables and also can change the values inside it. By using this approach, the potential problem of calling too many backend APIs after changing the pages was avoided. This is particularly evident after form submission, where all the data is sent to the backend and saved in the context, allowing the user to revisit the first step and see the previously entered information.

The context consists of an `activateStep` and a `setActivateStep` hook<sup>4</sup> to control the steps through the application. It has a `formData` and a `setFormData` hook, used to save all information from the form in the first step. With the function `updateFormData`, all fields can be updated properly. The `reviewID` and `setReviewID` hook are used in each endpoint call. ID is then generated on the backend and sent in response after successful form submission by the user. While user accounts do not exist in this initial prototype, `userId` and `setUserId` methods have been implemented. While these functions return static values in the current version, they are prepared to be updated in future versions of the application.

---

<sup>3</sup><https://react.dev/>

<sup>4</sup>React hook is the feature that allows to use lifecycle methods inside functional components.

## 5 Implementation

This paragraph describes each page of the application as well as the specifics of the implementation. All pages are created using Material UI<sup>5</sup> for visualization and React hook for the page logic. Additionally, open-source libraries were used to create this application. All libraries used in the project can be found in Table 1.

Name of the libraries			
Yup	React-Toastify	React-router	React-redux
React-hook-form	React-dropzone	Styled-components	React
Axios	mui/x-data-grid	mui/x-charts	mui/material
mui/icons-material	hookform/resolvers	emotion/styled	emotion/react
Vite	Papaparse		

Table 1: The list of all libraries

Figure 4 illustrates the workflow of the application. The blue start circle represents the starting point of the application, which is the Home page on the root URI. In this diagram, the ending point represents having completed the screening of a systematic review and it is realized as the last step in the stepper component. The orange boxes represent the separate pages, explained in the following chapters.

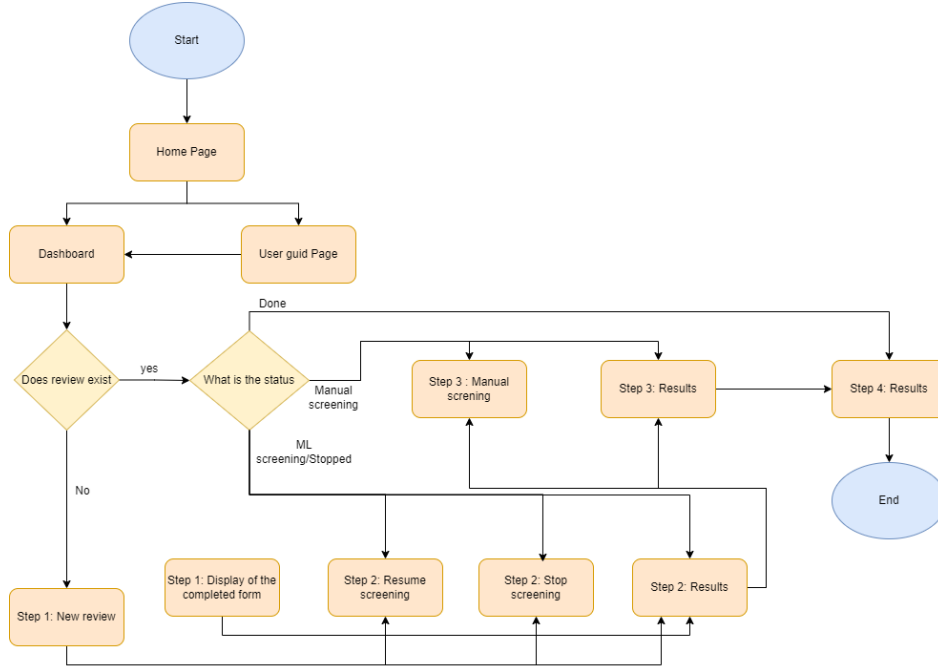


Figure 4: The workflow of frontend part of application

### 5.1 Home page

The home page is the first page that appears after starting the application (Figure 5). It contains a placeholder for a welcoming message and two buttons for easy navigation: one to switch to the

<sup>5</sup><https://mui.com/material-ui/>

dashboard and another to access the user guide. Since it is static and does not require data from the backend, it is developed using plain HTML and components of Material UI.

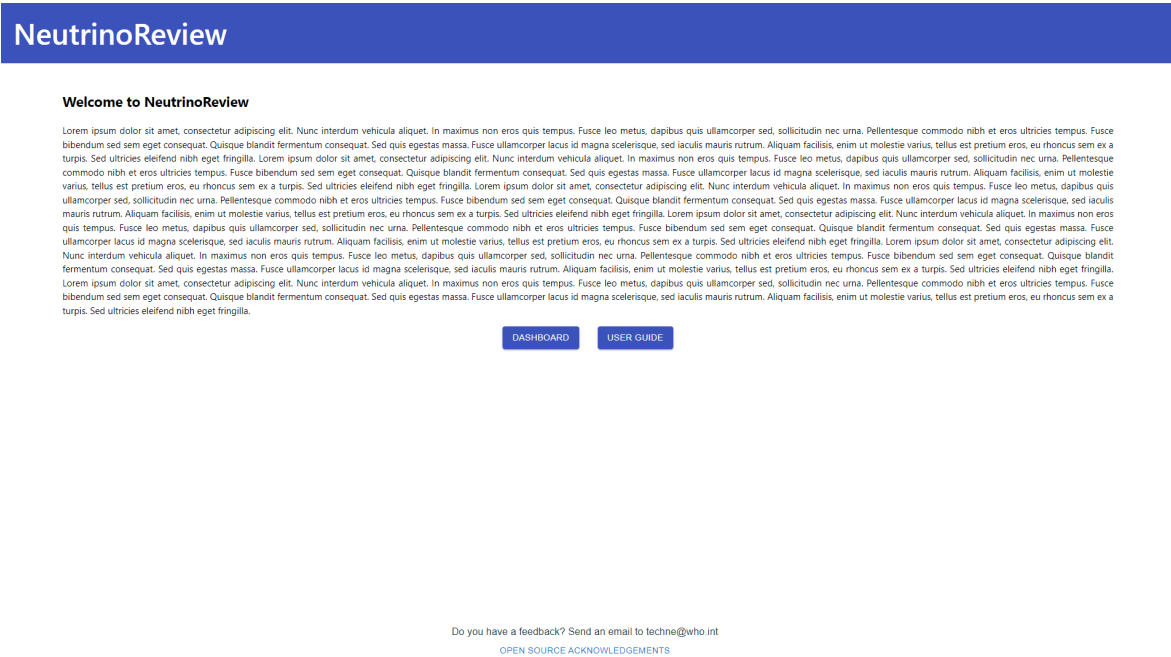


Figure 5: The home page

## 5.2 Dashboard page

The dashboard provides an overview of all created reviews, which are visually arranged in a table (Figure 7). Each record consists of the following fields: title, last modification, status, and progress. A record element provides the possibilities for record deletion, title editing, or navigation to the page where the user can continue the review.

The entire table design was made using the Material UI DataGrid component. After each action made in the table, the user gets a notification made using the Toastify library. If the user decides to edit the title of the review, a modal window will open that allows editing (Figure 6). By using the `useState` hook, containing all the records from the table, the edited record will be updated using the `setRows()` method, and will automatically be displayed in the table, and simultaneously sent to the backend to be saved in the database. It is important to emphasize that the opening of modal windows is also achieved using the `useState` hook since it is necessary to control the type of actions on this page.



Title

Edit the title below.

Viral Load: A systematic review for the ARIA project

CANCEL CONFIRM

Figure 6: The modal window to edit the title

Furthermore, if the user decides to delete the specified record, a new modal window will also be open where the deletion must be confirmed since the data is permanently deleted from the database. Finally, redirection to other pages, depending on the state of the review, is achieved using `useNavigate` and the context. Using this approach, it is possible to determine which step should be redirected. If the review status is *ML screening* or *Stopped*, it will proceed to step 2. If the status is *Manual screening*, it will move to step 3. Finally, if the status is *Done*, it will advance to step 4.

## NeutrinoReview

CREATE NEW REVIEW					
ID	Title	Last modification	State	Progress	Actions
1	Viral Load: A systematic review for the ARIA project	01/01/2023	ML screening	88	→ 🗑️
2	Efficacy of Vaccines in Preventing Influenza	01/01/2023	Done	90	→ 🗑️
3	Effectiveness of Antiviral Drugs in Treating COVID-19	01/01/2023	Stopped	90	→ 🗑️
4	Comparative Study of Diabetes Management Programs	01/01/2023	Done	90	→ 🗑️
5	Impact of Physical Activity on Cardiovascular Health	01/01/2023	Manual screening	90	→ 🗑️
6	Long-term Effects of Pediatric Asthma Treatments	01/01/2023	Manual screening	90	→ 🗑️
7	Long-term Effects of Pediatric Asthma Treatments	01/01/2023	Stopped	88	→ 🗑️
8	Long-term Effects of Pediatric Asthma Treatments	01/01/2023	ML screening	90	→ 🗑️
9	Long-term Effects of Pediatric Asthma Treatments	01/01/2023	Stopped	10	→ 🗑️
10	Long-term Effects of Pediatric Asthma Treatments	01/01/2023	Done	74	→ 🗑️
Rows per page: 10 1-10 of 15 < >					

Do you have a feedback? Send an email to [techn@who.int](mailto:techn@who.int)  
[OPEN SOURCE ACKNOWLEDGEMENTS](#)

Figure 7: The dashboard page

### 5.3 User guide page

The user guide page (Figure 8) provides comprehensive information to assist users in maximizing the benefits of using NeutrinoReview for TiAb screening. It consists of Accordion components from Material UI that automatically expand the card if the user clicks on a specific title. In order to realize the mentioned functionality, an array of objects was created, each of them consisting of an ID, title, and content. This page provides the possibility to navigate to the dashboard page if the user is familiar with all information.

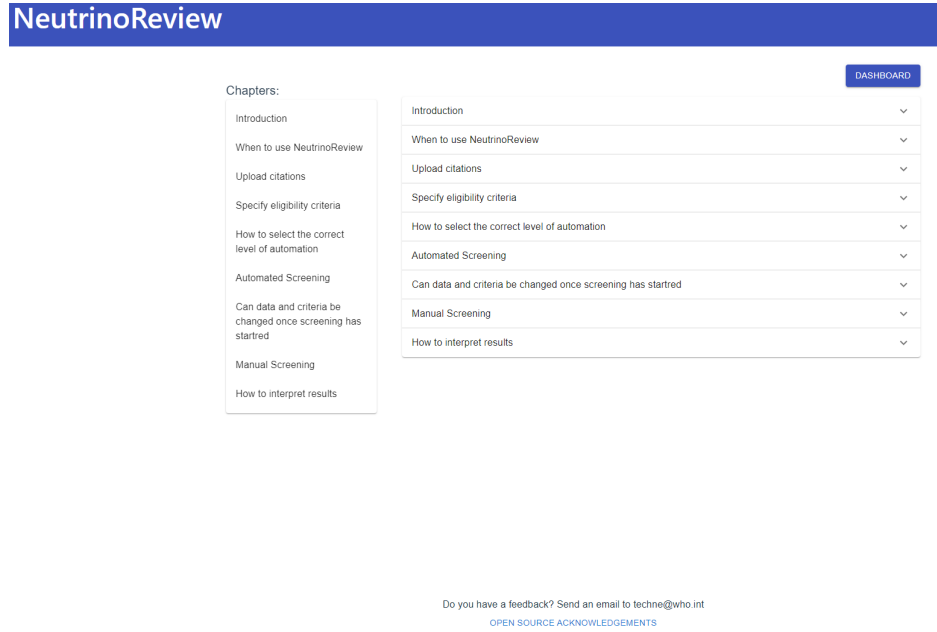


Figure 8: The user guide page

## 5.4 Mobile page

According to the specified requirements, it was defined that the first prototype of the application would be accessible only on desktop computers. The method used to accomplish that involves checking the screen size. If it is smaller than 768 pixels, a message is displayed to the user that "the current version of NeutrinoReview does not support mobile access.", and further use of the application is disabled.

## 5.5 Stepper

The stepper is the main component of this application. To execute screening, the user will be guided through 4 steps: *Define Specifications*, *LLM Screening*, *Manual Screening*, *Results*. In the frontend, this represents the most encompassing component. Moreover, by using the disabled property, it is ensured that the user can not access steps unless the current step is successfully completed. According to the specification, it is developed that the user can see only the current step and all the steps that have been successfully completed previously. This logic is implemented inside the `isStepDisabled()` method in the file `HorizontalGuid.jsx`. The movement from one step to another is realized with the help of indexes which are set in the variable `activateStep` which is defined in the `useContext` hook. In Table 2 can be seen which index corresponds to which page. Using the react provider, those variables can be accessed within all steps. It allows to pass data down through the component tree without having to explicitly pass props at every level.

Index	Step	Name of the Page
0	1	New review page
1	2	LLM screening
2	3	Manual screening
3	4	Results

Table 2: The index and steps in Stepper component

### 5.5.1 New review

After pressing the button "New review" on the dashboard, the user gets redirected to the first step where they can enter all relevant information (Figure 9). It consists of 4 fields. The first field refers to entering the title of the review, followed by the upload of the comma-separated values (CSV) file, defining the eligibility criteria, and selecting the desired level of automation. The entire page is made using the React-hook-form library, where only the field uploads CSV file is custom-made. All fields in the form are controlled using the FormControl component and they are also required to fulfill. Validation of the form is achieved using the Yup library, where custom text is defined for each field. Only after all the fields have been filled, the submission of the form is enabled and notification is displayed to the user, after which it is automatically redirected to the next step and starts the LLM screening. All values are saved in the context by calling the `updateFormData()` method since it is more efficient to save the data locally if the user returns to the previous step.

The most critical input to enable the LLM to properly execute automated screening are the eligibility criteria. It was created in such a way as to provide complete freedom to the user, with the requirement to create at least one record. It consists of three fields: label, inclusion criteria, and exclusion criteria. Fields are dynamically controlled with the `useEffect` hook. After the Add button is clicked, the field it refers to is automatically disabled using a flag in the `setDisabledRows()` method. However, it was decided to give the user the ability to delete the entire record, or to edit, whereby the entire record is disabled again after confirming the changes.

Name of your Review:

Upload Citations as CSV:

Please ensure that you only upload CSV files that include columns with the headers "Title," "Author," "Abstract," "Publication Year," "DOI," and "ISSN." Furthermore, ensure that the CSV file is already deduplicated. A detailed step-by-step guide on how to create a CSV file can be found in [the User Guide](#)

Click to browse or drag and drop your files

Eligibility Criteria:

To ensure that NeutrinoReview performs optimally, please provide clear, specific, and straightforward inclusion and exclusion criteria. Avoid combining several criteria into one. Also, consider that many specifications that are obvious to you as a human expert might not be for our AI system. This will help the model accurately understand and apply your criteria during the automated screening process. You can find detailed instructions and examples in [the User Guide](#).

	Included papers that...	Excluded papers that...	
<input type="text" value="Label"/>	<input type="text" value="Specify inclusion criteria..."/>	<input type="text" value="Specify exclusion criteria..."/>	<div> <div></div> <div></div> </div>

[+ ADD](#)

Level of Automation:

Please select one of the following options:

**Automation Level 1 (Max. Sensitivity):** The algorithm is designed to only automatically include and exclude those citations where it is able to make a very clear decision. This results in a higher number of papers to be screened by you as the human expert.

**Automation Level 2 (Balanced):** The model automatically includes and excludes most papers and only leaves a small number for human screening.

**Automation Level 3 (Full-Automation):** No manual labelling is necessary. The model classifies the whole dataset into include and exclude according to the highest probability. This option should only be chosen when time is critical and missing few relevant papers can be accepted.

Select Automation Level:

☐ Automation Level 1 (Max. Sensitivity)  
☐ Automation Level 2 (Balanced)  
☐ Automation Level 3 (Full-automated screening)

[START AUTOMATED SCREENING](#)

Figure 9: Define Specifications Page: The first step in the creation of a new review

### 5.5.2 LLM screening

After providing all relevant input data and clicking on the "Start automated screening" button, the user gets redirected to the LLM page on step 2 (Figure 10). It consists of two intermediate steps: AI screening and displaying results. Firstly, it shows a custom progress bar, buttons for stopping/resuming screening, and information about the current state (number of documents, remaining time). Dynamically, the frontend receives a new response every second. This response consists of the remaining time, the number of screened documents, progress, the states, and results used in further steps. All those values are saved into the useState hook. The LLM screening process can also be stopped by scanning activated by pressing the button, where the string *stop* or *resume* is sent to the backend and the useState hook is set to true/false. After the progress reaches 100, a new component that displays a graph in the form of a pie chart is automatically rendered (Figure 11).

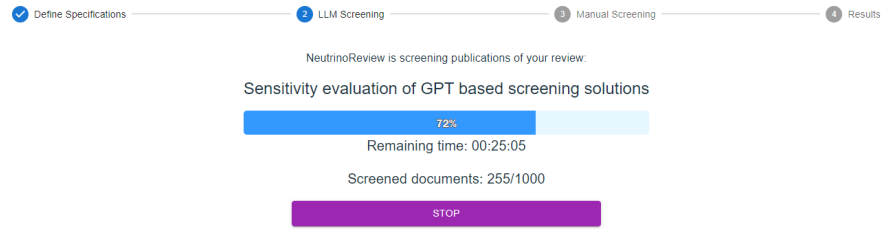


Figure 10: LLM Screening Page: The second step in creation of new review

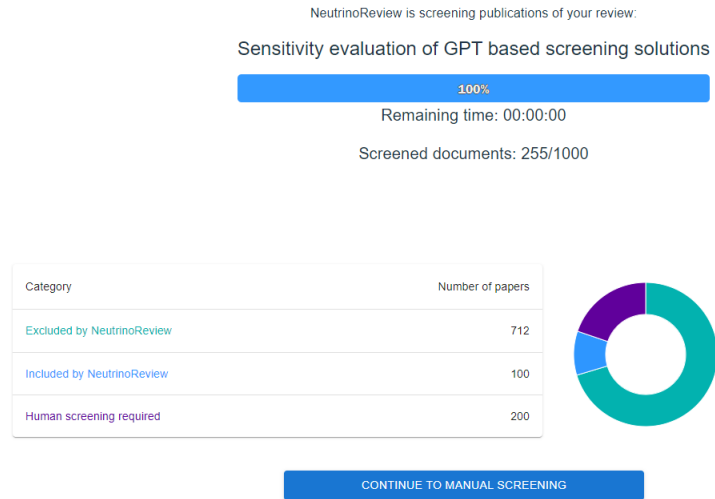


Figure 11: The graph with results after LLM screening process

### 5.5.3 Manual screening

Manual screening is the third step in the review process (Figure 12). Like in the previous step, it consists of two intermediate steps: manual screening and display results. The user is shown articles that were not classified in the LLM screening.

Each record in the table is accompanied by two buttons to select its screening decision ("include" or "exclude"). To remind the user to strictly follow the predefined eligibility criteria, these are displayed on the left side of the page. Furthermore, the user has the possibility of viewing documents that have previously been manually screened using the back and next buttons are displayed on the left. Furthermore, the user has the possibility to change previous decisions. If "include" is selected for the classification, the endpoint for the decision will be automatically called. On the other side, in the case of "exclude", a new modal window will be opened with a display of exclusion criteria categories and the possibility of selecting a new category. The logic in modal windows is developed using a separate useState hook for custom categories from the backend. After the decision is made, the next

article is automatically retrieved. If `switch\_to\_results: true` is returned from the backend, the components on this page will be replaced with a graph showing the results.

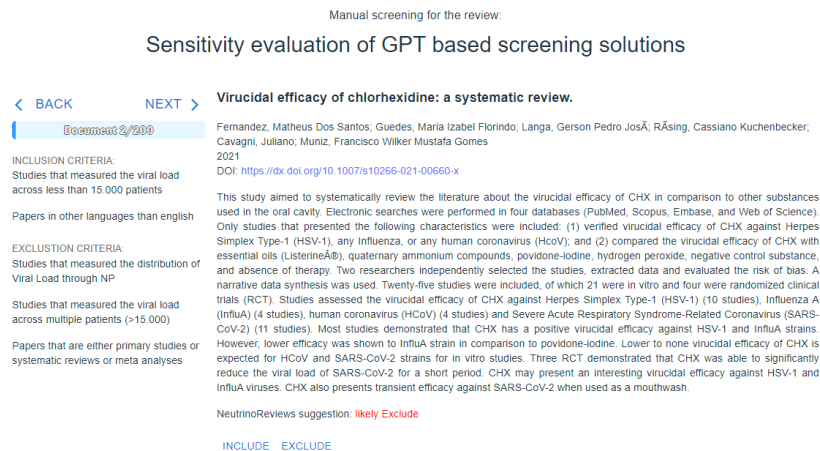


Figure 12: Manual Screening Page: The third step of screening literature with NeutrinoReview

## 5.5.4 Results

The final step is the result page, which shows a systematized overview of the previous two steps (Figure 13, 14). This page consists of 4 components.

The first component is the PRISMA diagram (Preferred Reporting Items for Systematic reviews and Meta-Analyses)[9]. It shows in detail the revision of the literature. It is developed with Canvas JS where each element is drawn using JavaScript and it was necessary to specify the exact position (in pixels) of the element due to the custom diagram. Secondly, the same page shows the graphic presentation of the same data using the same Material UI component as the previous two steps. Using the DataGrid component from Material UI, citations are displayed with the include/exclude tags and reasons. Finally, the page enables the possibility to download the table in the form of a CSV file. This method is developed using the Blob (Binary Large Object) interface.

PRISMA Diagram:

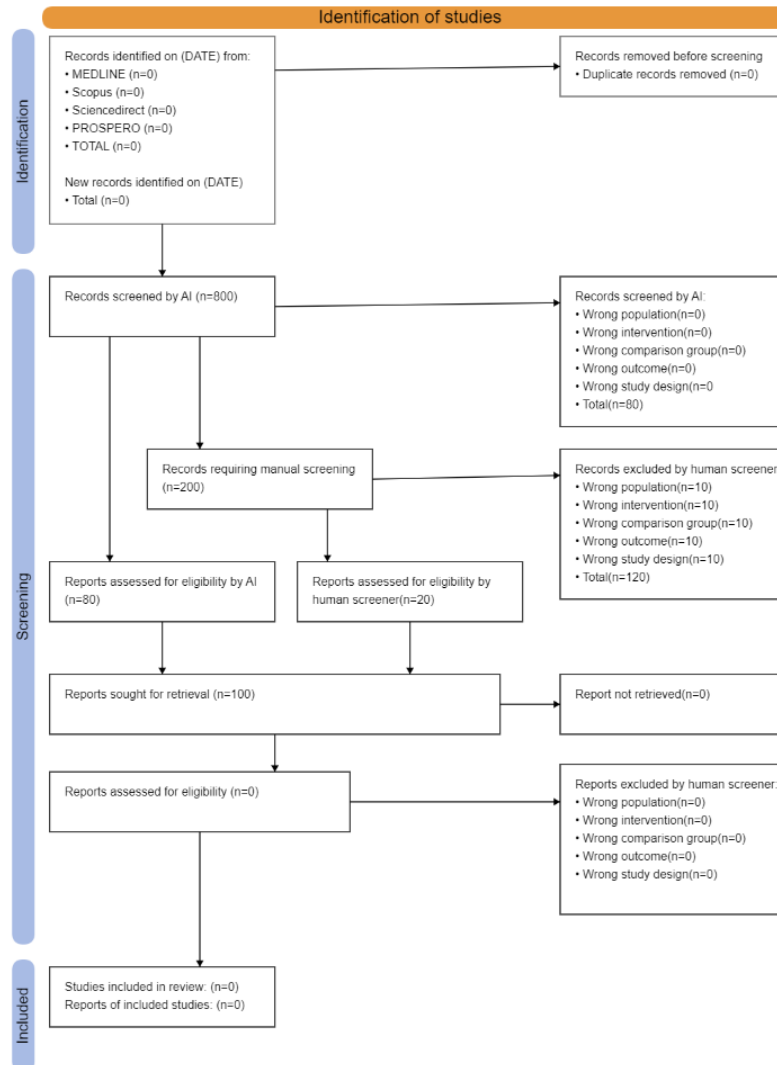
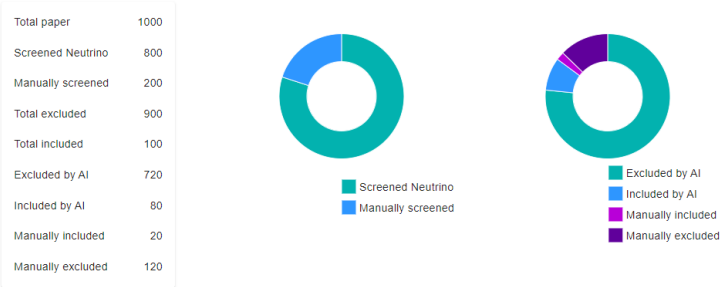


Figure 13: The PRISMA diagram from result page



List of citations:

ID	Tag	Title	Authors	Exclusion reason	Screener
1	Exclude	Efficacy of Pre-Procedural Mouthwashes against SARS-CoV-2: A Systematic Review of Randomized Controlled Trial	Garcia-Sanchez, Alvaro, PeÅsa-Cardelles, Juan-Francisco, Ruiz, Steve, Robles, Flor, Ordenez-Fernandez, Esther, Salgado-Peralvo, Angel-OrlÅn, Balloch, James, Simon, Jacob C.	Lorem ipsum dolor, Lorem ipsum dolor	Neurino Review
2	Exclude	Virucidal efficacy of chlorhexidine: a systematic review.	Fernandez, Matheus Dos Santos; Guedes, Maria Izabel Florindo; Langa, Gerson Pedro JosÅ; RÅfising, Cassiano Kuchenbecker, Cavagni, Juliano; Muniz, Francisco Wilker Mustafa Gomes	Lorem ipsum dolor, Lorem ipsum dolor	Neurino Review
3	Include	Cross-Species Translation of Correlates of Protection for COVID-19 Vaccine Candidates Using Quantitative Tools	Fernandez, Matheus Dos Santos; Guedes, Maria Izabel Florindo; Langa, Gerson Pedro JosÅ; RÅfising, Cassiano Kuchenbecker, Cavagni, Juliano; Muniz, Francisco Wilker Mustafa Gomes		Mario N.
Rows per page: 10 1-3 of 3 < >					

Download CSV:

Download a list of all citations with comprehensive annotations regarding the screening results. For a thorough explanation on how to interpret these results, please refer to [the User Guide](#).

DOWNLOAD AS .CSV

Figure 14: The rest of result page



## 6 Conclusion

In conclusion, this report presents the frontend web application for NewutrinoReview which is part of the CaiMIRA WP4. The implemented web application guides the user through the TiAb screening phase of an SR. By following the four-step process, the user is able to provide all required information, consult an LLM for automating the major workload, manually screen those papers where the LLM was not able to make a clear decision and see the results as a last step. Using this application, we assume that we will reduce the time which user needs for SR.

The outcome of this summer student project is the frontend for the first prototype for NeutrinoReview. It is limited to the TiAb screening phase and provides LLM-based automation of this labor-intense task. However, as an initial prototype, there is room for enhancement in future versions. Firstly, this initial prototype is designed for desktop devices, but future versions could be expanded to include tablet compatibility. Furthermore, the option to select from several LLM models could be given to the user, allowing them to choose based on cost and performance variations. Also, features like keyword highlighting, which are available at tools like Covidence<sup>6</sup> and also the possibility to create user accounts are left for future work. Furthermore, a login should be implemented as well as the possibility to collaborate.

During these eight weeks, I gained extensive knowledge about CERN and the SR process. My frontend development skills improved significantly, largely due to daily interactions with the CAiMIRA team. Notably, our team approached the development of the application in a highly structured manner, dedicating substantial effort to refining the eligibility criteria table and the PRISMA diagram. Both components required iterative discussions to identify the best development solutions. Ultimately, problems with the PRISMA diagram were solved using Canvas JS while for the design of the eligibility criteria table was decided to have three separate cards that have options to delete, edit and add. All records are dynamically controlled using the useRef hook.

In summary, the developed UI provides a foundation for the first version of the NeutrinoReview tool, enabling researchers across various domains to significantly reduce their workload during the TiAb screening stage of systematic reviews. This project is open source and it is available on CERN GitLab of CAiMIRA project (<https://gitlab.cern.ch/caimira/neutrinoreview>)

---

<sup>6</sup><https://www.covidence.org/>

## References

- [1] K. Coleman, S. Norris, A. Weston, K. Grimmer-Somers, S. Hillier, T. Merlin, P. Middleton, R. Tooher, and J. Salisbury, “NHMRC Additional Levels of Evidence and Grades for Recommendations for Developers of Guidelines,” 2009.
- [2] M. Bartholomew, “James lind’s treatise of the scurvy (1753),” *Postgraduate Medical Journal*, vol. 78, no. 925, pp. 695–696, 2002.
- [3] I. G. Needleman, “A guide to systematic reviews,” *Journal of clinical periodontology*, vol. 29, pp. 6–9, 2002.
- [4] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.
- [5] M. R. Demetres, D. N. Wright, A. Hickner, C. Jedlicka, and D. Delgado, “A decade of systematic reviews: an assessment of weill cornell medicine’s systematic review service,” *Journal of the Medical Library Association: JMLA*, vol. 111, no. 3, p. 728, 2023.
- [6] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.
- [7] A. M. Methley, S. Campbell, C. Chew-Graham, R. McNally, and S. Cheraghi-Sohi, “Pico, picos and spider: a comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews,” *BMC health services research*, vol. 14, no. 1, pp. 1–10, 2014.
- [8] A. Henriques, N. Mounet, L. Aleixo, P. Elson, J. Devine, G. Azzopardi, M. Andreini, M. Rognlien, N. Tarocco, and J. Tang, “Modelling airborne transmission of sars-cov-2 using cara: risk assessment for enclosed spaces,” *Interface Focus*, vol. 12, no. 2, p. 20210076, 2022.
- [9] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, P. Group, *et al.*, “Preferred reporting items for systematic reviews and meta-analyses: the prisma statement,” *International journal of surgery*, vol. 8, no. 5, pp. 336–341, 2010.