Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Design and Implementation of a Platform for Discovering and Sharing AI Planning Software

Saurabh Vijay Adhau

Course of Study: MSc Computer Science

Examiner: Dr. Ilche Georgievski

Supervisor: Dr. Ilche Georgievski

Commenced: April 8, 2024

Completed: October 8, 2024

Abstract

One of the branch of Artificial Intelligence is AI planning and over the past few years, there has been significant research in the AI planning field which led to development of multiple techniques, algorithms and tools. Yet while AI planning techniques have been moving forward with research, the lack of a central repository for sharing AI planning software may have hindered their broader adoption. Resources are currently scattered across web-pages, research papers and books and this situation complicates finding appropriate AI planning software as per requirements by people who are from the field and even more difficult for new users of AI planning. The lack of standardisation and accessibility of planning software brings the field in a situation that hinders applicability and reproducibility. The goal of this master's thesis is to design and implement a single platform that provides access to multiple AI planning software artefacts. The platform is designed with a modular architecture with different functionalities as a service which ensures seamless integration between frontend and backend. This platform provides users with an intuitive UI with features to discover, filter and sort. The solution is evaluated through user studies and it highlights positive feedback on platform's design and features while it also guides on areas of improvement for overall user experience and widespread adoption.

Keywords: Artificial Intelligence, AI Planning

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 22

AIMe Artificial Intelligence in bioMedical research. 22

API Application Programming Interface. 37

DB Database. 29

FR Functional Requirement. 26

GUI Graphical User Interface. 38

HTN Hierarchical Task Network. 18

ML Machine Learning. 22

NFR Non-Functional Requirement. 27

OpenML Open Machine Learning. 22

UI User Interface. 41

1 Introduction

In artificial intelligence, AI planning is the task of automatically producing one or more plans to accomplish a goal [1]. The main task of AI planning algorithms is creating a series of steps whose execution leads from an initial state to the goal. This applies to many fields from robotics, autonomous systems and logistics scheduling that is in short whenever we need real-time planning or decision making in complex environments [2]. In autonomous driving, the car has to figure out the best path it should take, when it should stop and how best to avoid potential obstacles at every fraction of a second [3].

AI planning has the potential to be used in real-world applications and it is playing an important role in a variety of domains, including healthcare, space exploration, cloud computing [4] and smart buildings [5][6]. AI planning is also playing an important role in medical IT systems, where it can assist healthcare professionals in creating, reviewing, and adjusting standardized care plans [7]. It also enables the customization of care plans to meet the unique needs of individual patients and adapt to different healthcare environments [7]. In the area of space exploration, missions are often highly dependent on planning systems that can autonomously organize many tasks and execute them without human interventions [8]. Applications of AI planning can be used to solve problems across a wide range of domains that involve long-range decision making in the presence of uncertainties and constraints [2]. Its effectiveness in these domains show its importance and potential for future research and innovations.

One big problem in the AI planning community is that collaboration as a whole is hampered by the absence of a common ground for sharing resources, algorithms and tools [9][10]. Today, AI planning software are scattered throughout a numerous of individual websites, research papers and repositories, limiting researchers as well as practitioners to easily being able to keep pace with the current technologies in this area. Due to the fragmented nature, somewhere research in this field is also slowed down. A shared, centralized platform hopefully could help in this field by making the information available and easy to find to most people, with the ability to further contribute to it.

In this thesis, we go through the overview of AI planning. With respect to the research question, we perform research in AI planning field and related fields and find out it's contributions and open challenges. Later, we come up with the design of the solution, it's implementation and perform a user study for evaluation of the solution to find usability of the solution.

1.1 Problem Statement

Not only the issue of fragmented information, but also many AI planning tools have a little longer learning curve [11]. Provided there are a plenty of AI planning algorithms available and numerous software implementations, most of them are slightly time consuming to use and such software require deep technical expertise. For instance, there might be a AI planning software available with PDDL [12] planning language and in that scenario where user is comfortable with a different planning language, user must go back and forth during the search to find the software as per the requirements and individual expertise [13][14]. This forms a major hurdle for user groups without much knowledge on AI to engage with these technologies, thereby constraining their widespread adoptions.

The existing metadata that would enable users to quickly evaluate the functionality and suitability of the AI planning tools available is not only incomplete but also lacks usability. The absence of detailed searchable metadata makes it even more difficult for users, which increases the overall barrier when trying to compare tools and in turn it results into limited adoption of AI planning tools.

Explaining the above issue, for example it is hard to say which type of planning tool that meets the needs of a given task. Different AI planning tools are optimized for a different types of planning classes such as classical planning, temporal planning and others [15]. Also, those tools can be of different planning types such as Solving, Monitoring and so on [16]. Identifying which planning software supports specific classes and functionality types can be a time-consuming process due to the scattered availability of documentation [14] and inconsistent metadata. This understanding only comes after doing a through research and with sufficient understand of the domain knowledge. And it gets difficult for non-experts to understand which tool to use, and making mistakes during the discovery of AI planning software could happen. This lack of standardization and user knowledge leads to the available resources not being discovered and used.

Users find it difficult to discover, access and evaluate the right tools for their particular use case [16]. The lack of consolidated data not only take more time for understanding as explained before but also leads to inefficiencies in selecting and setting up the right software. For example, when the metadata for AI planning software is not easily available, users might choose a tool based on their requirements but overlook platform compatibility (Windows, Linux, Mac, etc.), which is not clearly visible and may be present in code repositories files. As a result, they would need to conduct an additional search to finalize the right software. The lack of a unified repository or any system providing an end-to-end understanding about available AI planning tools showcasing features, can and help users based on the requirement of selecting the perfect tool, is holding back the widespread adoption of AI planning.

1.2 Research Question

The research question of this project is: How can we design and develop a platform for AI planning software that fosters widespread adoption of AI planning? This question is important as it addresses the key issues and challenges which are limiting the accessibility and usability of AI planning software. A good answer to this question would be a roadmap on how to build a platform that not only consolidates AI planning tools in one place but also makes these tools user-friendly, flexible and usable by a wider group of users[17].

The first aspect of answering this research question is to determine how to gather and compile the meta-data of AI planning tools into a central repository. This involves studying the current AI planning software ecosystem, consolidating all commonly used tools and figuring out an optimal way to classify them as per functionalities, and their purpose. The platform also uses metadata to record important information about each planning software, including key attributes such as documentation, the implementation language, the operating system, planning language, planning types and , making it easier for users to find the tools most suitable for their desired task. We ensure that the information is detailed and accessible and it is an important step to foster adoption of the platform.

The second part of this research question covers the design concerns related to platform architecture, focusing on key principles such as modularity, maintainability, and portability. By ensuring a clear organization of components based on their responsibilities, the architecture enhances maintainability and scalability, allowing the platform to evolve without significant disruptions [18]. Additionally, the platform's portability ensures that all parts of the system use consistent configurations, eliminating the need for separate installations or environment-specific customizations [19]. The platform design aligns with the FAIR [20] principles by having the properties of being Findable, Accessible, Interoperable and Reusable which means it allows for easy discovery of AI planning software and their metadata, accessibility via standard interfaces, compatibility against common formats, and modular structure for reusability [20].

The third part in the response to the research question is finding out which are the different user groups who will use the platform. This could mean AI researchers, industry professionals or students with differing levels of experience. For this reason, the platform must serve all those groups and provide from user-friendly interfaces so that non-experts can use them to many advanced configuration options for experts. It also needs to encourage collaboration by allowing users to share their knowledge and add software details collaboratively and help establish a community of practice around AI planning. The entire system is intended to foster research and development of advances in AI planning, through collaboration and knowledge sharing via common ground.

Evaluating the usability of the platform is an another important aspect while answering the research question. This includes user studies to evaluate how an intuitive and useful the platform is addressing different types of users. This helps us to get feedback from real users by which we can interpret the user satisfaction. The outcomes of these evaluations informs how the platform can be improved in terms of design and functionality to maximise its impact on encouraging adoption across different domains for use with AI planning tools.

2 Background Information

We go through the fundamentals of AI planning, discuss the existing techniques and AI planning types.

2.1 Overview of AI Planning

AI Planning is an area of artificial intelligence which focuses in the design and generation of plans, or sequences of actions, to determine the action at each step in order to achieve the goal. At a very basic level, AI planning comes down to finding steps or actions that can lead from some initial state of the world to a goal state [21]. This ability to produce action sequences makes AI planning especially useful for complex, changing environments that require autonomous decision-making [22]. The AI problems in the non-trivial plan are more complex where they involved multiple steps, constraints and goals need to be searched within reasonable time.

At the core of AI planning are the basic ideas of states, actions and goals [23]. A state captures certain variables and conditions of the world at a key point in time. Action is an operation that by changing a specific feature of the state cause the movement of one state to another. A goal on the other hand, is an end condition that a system should ideally get to, usually persisted in specific state variables [24]. In practice, an AI planner works by combing through potential sequences of actions (commonly called a plan) that move from the start state to the end goal state. This search involves moving through the state space, which is nothing but a theoretical space where every possible state and transitions are mapped.

The most difficult part of AI planning is controlling the combinatorial explosion in the state space [25]. The number of different plans in graphs grows exponentially with the number of states and actions to consider and as such it becomes tremendously hard to find the best one from a computational point of view. This problem is most pronounced in real-time or dynamic environments, where decisions must be made rapidly [26]. To address this problem, most AI planning algorithms use heuristics: sets of rules or techniques that help the search proceed faster by being more focused on potentially useful actions. In general, these improvised heuristics are derived from domain-specific knowledge or general rules about the problem.

To recap, AI planning is a crucial capability in autonomous decision-making within scenarios necessitating intricate, multi-step workflows [2]. AI planners solve real-world problems by defining the world in terms of states, actions, and goals and searching through possible sequences of actions. But the high-complexity of most AI planning tasks makes it computationally expensive to perform without sophisticated algorithms and heuristics.

2.2 Existing Techniques and Types

AI planning includes a variety of problems, ranging from the simple pathfinding task in static world to complex multi-agent systems undergoing actions in dynamic and uncertain environment [27]. Following are the different AI planning classes [24]:

- **Classical Planning**: This class deals with deterministic actions in a fully observable environment. The planner treats each action as though it will result in a particular outcome, and that this outcome is predictable.
- **Numerical Planning**: The planning that includes numerical constraints and variables. It generalizes traditional planning approaches to metrics like resource boundedness or objective-based optimization, such as minimizing time or cost.
- **Temporal Planning**: This planning class involves actions that take time and the scheduling of actions to occur in time. It is important in situations where the order and overlap of actions are both significant.
- **Hierarchical Task Network (HTN) Planning**: This planning approach breaks down tasks into smaller, manageable sub-tasks. The planner decomposes complex problems into simpler sub-problems using a hierarchical solution technique.
- **Probabilistic Planning**: This class deals with uncertainty about the results of actions. In contrast to deterministic approaches, it considers a wider range of possible outcomes based on environments where actions have probabilistic effects.
- **Conformant Planning**: This class deals with environments where the initial state is partially unknown or where the outcomes of actions are not fully observable. A planner generates a plan that accounts for uncertainty and functions despite the unpredictability of the environment.

AI planning also includes different types of planning functionalities, adapted to the specific problems that these types of planners can solve. These include the Solving and Monitoring [16]. Problem solving is the process of creating a sequence of actions or decisions that lead to achieving the goal. This may involve some amount of exploration across the state space and evaluation of different candidate strategies to settle on the best course. In contrast, monitoring is concerned with tracking and observing the environment while the plan is getting executed. and it ensures that it follows the predefined conditions, goals, and adjusting the actions if necessary, either due some deviation or a changing behaviours in the environment also called as re-planning.

In addition to the above ones, there are several other planning functionality types suitable for different requirements and few of them are listed below [16]:

- **Learning**: By integrating lessons from the history, this feature helps in the planning process. Machine learning techniques are used to refine future plans based on the outcomes from previous attempts.
- **Translation**: Converts high-level planning problems into other types of computational problems, such as optimization, constraint satisfaction, or satisfiability problems. Through translation of the problem, planners can use dedicated algorithms that are more efficient and adaptable at solving specific aspects of the task.

- **Plan Validation**: Ensures that the plans are accurate, feasible, and follow all constraints provided either initially or during execution.
- **Fault Tolerance**: Deals with unexpected events or contingencies that occur when the plan is running, but an action does not go according to plan due to factors beyond control.
- **Execution**: Refers to implementing the planned actions to realize the plan in reality.

All of these planning functionality types address particular problems, and the type used is often determined by the demands of the problem concerned. In dynamic environments, it might make sense to optimize on Learning when previous experiences make future outcomes cheaper and faster than Planning, while executing in real-world like environments requires that complex plans remain correct and executable would best fit Plan Validation type. These different types are generally used to select the most appropriate planning service in terms of nature or functionality, which depends on how complex the problem is, what area it falls into and how much uncertainty is present.

3 Related Work

We go through the current research (with respect to our research question) done in the field of AI planning as well as from the related fields like Machine Learning and AI in Medical Indutry. We also do the critical analysis by looking at the contribution, open challenges and analyzing to what extend is that answering our research question.

3.1 Within AI Planning Field

A significant contribution towards an effort in consolidating AI planning knowledge via the creation of open platforms for sharing AI planning problems is Planning.Wiki [28]. It is an important resource as it acts as a consolidated repository of planning-related information, including definitions, algorithms, and tools. Mostly, it is an educational platform that has a theoretical description of popular planning systems (for example Fast Forward [29] and Fast Downward [30]). It can be perceived as a great resource for learning about AI planning concepts, but not suited to deliver metadata or offer specific features that would help users to find or compare different AI planning software. Moreover, it has limited number AI planning software on their platform. This thesis consolidates metadata for various AI planning tools, allowing users to search, sort, and filter available software in a unified repository.

The IPC [31] has been instrumental in advancing AI planning techniques by benchmarking planners on standardized domains. It creates a competition-driven environment that drives the development of new algorithms, but really focuses on ensuring that the performance of planners is measured. The International Conference on Automated Planning and Scheduling (ICAPS) [32], where IPC is hosted, primarily serves as an academic forum and does not offer a platform dedicated to centralized tool discovery or metadata beyond competition results. A new website is created for the International Planning Competitions (IPC) every year. Its reason is each year's competition focuses on specific domains, tracks, and organizers, which requires creating a unique website. Due to this nature, it is having fragmented information over different websites. This thesis addresses the fragmented nature and comes up with a single repository for AI planning software artifacts.

GitHub [33] hosts a wide array of open-source AI planning tools and software implementations, but it is not exclusively dedicated to AI planning field. It is a generalized code sharing repository and when it comes to meta data format there is no standardization and users can have their own different styles. Users may face challenges in identifying suitable tools due to lack of standardized metadata or particular search or sort features. Documentation can vary widely in quality, making it difficult for non-experts to understand or evaluate the software's meta data quickly. This thesis addresses the gap by unifying metadata in a standardized format, allowing users to search and filter based on specific attributes, thereby reducing the burden of manually going through scattered resources.

3.2 Broader Overview

And one of the biggest thing that has happened in the past decade is the research in Machine Learning (ML), Artificial Intelligence (AI) domain which gave rise to a few platforms to foster collaboration and sharing resources in artificial intelligence and machine learning. It enables researchers to share their datasets, algorithms and tools publicly, keeping all necessary information open. These include platforms such as Open Machine Learning (OpenML) [34] and Artificial Intelligence in bioMedical research (AIMe) [35], which have proven to be successful examples in the overall AI and ML community with their emphasis on transparency, reproducibility, collaboration across research communities. The design of these platform provides lessons from the point of view of organizing, accessing and reusing shared resources that are informative in developing a similar platform for AI planning software

OpenML [34] is a popular platform for posting machine learning datasets, models and experiments. It also permits researchers to upload experiments and other sorts of materials, thus enabling a community to perform large-scale comparisons or meta-analyses. OpenML is an open science platform that has one of its strengths in the fact that it connects several tools for machine learning, including Weka, KNIME and R. These integrations allow users to execute experiments easily and to share their results without the need for specific hardware configurations. OpenML is an example of how a shared platform can simplify the sharing of resources and improve collaboration by offering standardized formats for data and results as we are proposing here for the AI planning community.

AIMe (Artificial Intelligence in Medicine) [35] is a platform that has supported the recent progress of AI research on healthcare to publish tools, datasets and algorithms of medical application. It is specifically tailored to the healthcare domain. Over the past few years, a very active community around AI in medicine grew up around meetings for researchers to report findings and collaborate on studies. While one of the aims is to facilitate scientific information exchange, AIMe also tackle specific problems relating AI in healthcare: the necessity for interpretability and transparency in AI models. AIMe is a relevant case not only for the key lessons learned about AI planning within it but also as to how an accessible and sustainable software platform can make its reach by a wider audience and easy for adoption.

OpenML [34] and AIMe [35] also contain a common message on integration and inter-operability without which shared platforms are not that great. OpenMLshows how interoperability can break down barriers to the submission and availability of resources in the most common machine learning tools. One such extension of this kind is AIMe, a platform with a specific focus on medicine allowing it to tailor the tools and datasets their users can access in order perform targeted research. These platforms illustrate that in order for an AI planning platform to be successful, it has to focus on good integration of meta-data of current planning tools.

A key strength of platforms such as OpenML [34] and AIMe [35] is reusability and reproducibility. For example, OpenML enables queries of existing datasets and models, re-execution of experiments and comparison to the results of thousands other studies. This not only promotes transparency but also a lot of reusability, since researchers are able to work from the results from other papers rather than starting over. Such principles are especially important in AI planning for which the sharing of reproducible planning models and experiments can substantially contribute to progress within the field.

OpenML [34] and AIMe [35] both platforms have the facility of sharing and collaboration. They also have the capability of adding new research information into the platform. But they are into related domain and not AI planning. The information of AI planning tools is scattered across multiple places. Still the standards of sharing and collaborations by these platform as an ideas is beneficial as by collaboration not only researches can add their information but it also gets available on a unified platform with wider audience.

4 Platform Design

We come up with objectives of our platform and discuss the current limitations of existing AI planning platforms. We define the functional and non-functional reqirements of the platform, design the system architecture, decide the data models required, and also come up with blueprints called as web sketches for the platform. Lastly, we also compute a score for each AI planning software which tells us about the coverage of meta data absed on importance.

4.1 Objectives of the Platform

The objectives of the platform are derived from an analysis of the current challenges in the AI planning field and the limitations of existing platforms. The main objective is to create a centralized repository for AI planning software. The objectives are designed to better address the needs of key user groups, including students, researchers, and AI practitioners, ensuring the platform serves its purposes effectively.

1. **Develop a Unified Platform for AI Planning Software**: The primary objective is to design and create a platform to aggregate individual AI planning tools' information from scattered locations into one system. This allows users to access a range of AI planning software easily, giving researchers, practitioners, and students the chance to find and use suited planning software for specific tasks.
2. **Support Collaboration and Knowledge Sharing**: The platform should foster collaboration among AI planning researchers and practitioners. Users from the AI planning community will be able to share their knowledge and developed AI planning software on the platform, benefiting the entire community.
3. **Promote Widespread Adoption Across Different Users**: Another objective is to build the platform with different user groups in mind, such as researchers, practitioners, students, and anyone involved in AI planning. The key is to keep the platform simple enough to support a variety of users.
4. **Evaluate Platform Usability through a User Study**: A critical objective is conducting a user study to validate the platform's usability and effectiveness. This will include feedback from beginner to expert users, assessing how well the platform meets their needs, and making iterative improvements based on real-world usage.

4.2 Requirements

The requirements for this platform were derived from a critical analysis of existing platforms in the AI planning field, including platforms like Planning.Wiki [28], International Planning Competitions (IPC) [31], and GitHub [33] repositories. By examining the limitations of these platforms, key functional and non-functional requirements emerged. The analysis revealed the need for features like search, filter, and sort functionalities to improve accessibility. The primary user group involved in this process are students, researchers, AI practitioners. By considering this user group, who needs easy access to AI planning software for research, development influenced the formulation of these requirements to ensure that the platform meets different user needs.

4.2.1 Insights from Existing Efforts

1. **Importance of Standardization**: Our analysis of present platforms like OpenML [34] and AIMe [35] highlights the importance of better standardization across platforms. A common AI planning software platform would dramatically reduce the complexity of browsing through different websites, research papers, or source code repositories.
2. **Challenges in Design based on User Groups**: Another key takeaway comes from user-centric design. While technical users may have familiarity with setting up AI tools, many potential users do not. It is crucial to ensure that the platform has an intuitive design so it is easy to use. Key features should include intuitive search and filter options, as well as tutorials and user guides to help users select and use planning tools.
3. **Balancing Complexity with Usability**: Another lesson from the previous analysis is the need to balance technical information with ease of use. For example, OpenML [34] has many nested features, which can hinder efficient use. The conclusion is that a simpler platform is easier to use, ensuring users can engage with it effectively.

4.2.2 Functional Requirements

1. **Functional Requirement (FR)1: Search Functionality**:
   - Platform users should be able to perform a global search for AI planning software via a search bar.
   - The results of the search should return AI planning software with the matching keyword from the search bar.
2. **FR2: Sorting Functionality**:
   - Users should be able to sort the complete list of software listed on the platform using the sort button.
   - The sorting feature provides multiple options to sort the data in both ascending and descending order.
3. **FR3: Filtering Functionality**:

- Users should be able to filter AI planning tools based on various parameters.
- Upon filtering, only those tools which match the respective filter criteria should be displayed.

4. **FR4: Card Interaction and Detail View**:
   - Users should be able to navigate from the card view, which displays important attributes of AI planning software, such as the year, abstract, and other key metadata, to a detailed view page. Each card provides a concise overview of these attributes, and clicking on a card will take users to a details view page.
   - The details view page displays complete information about the AI planning software in a tabular format.
5. **FR5: Add Functionality**:
   - Users should be able to add details of new AI planning software using the ADD button.
   - Users are redirected to the Add page, where they can insert data via various input fields.
6. **FR6: Support and Contact Feature**:
   - Users are able to contact the admin via a dedicated email address.
7. **FR7: Responsive Design**:
   - The platform should be accessible and usable across various screen sizes, including desktops, laptops, and tablets.

4.2.3 Non-Functional Requirements

1. **Non-Functional Requirement (NFR)1: Maintainability**: Code should be maintained in such a way that it can be easily understood. It should be possible to modify and enhance the initial code if required.
2. **NFR2: Usability**: Users should be able to easily understand the user interface. A user-friendly platform helps in promoting wider adoption.
3. **NFR3: Portability**: The platform should be able to work on different systems and machines irrespective of operating systems and installed software. This makes the platform system-independent.
4. **NFR4: Separation of Concerns**: The different functionalities of the platform should be separated into different layers based on their respective aspects. This is essential to ensure modularity, maintainability, and scalability.
5. **NFR5: Reusability**: The components of the code should be written in such a way that they can be easily re-used, avoiding code duplication. This also aligns with FAIR [20] principles R.

The platform design could also align with FAIR [20] principles where metadata is findable, accessible via endpoints, interoperable by using standard data formats and reusable as discussed before.

4.3 System Architecture

Platform Architecture

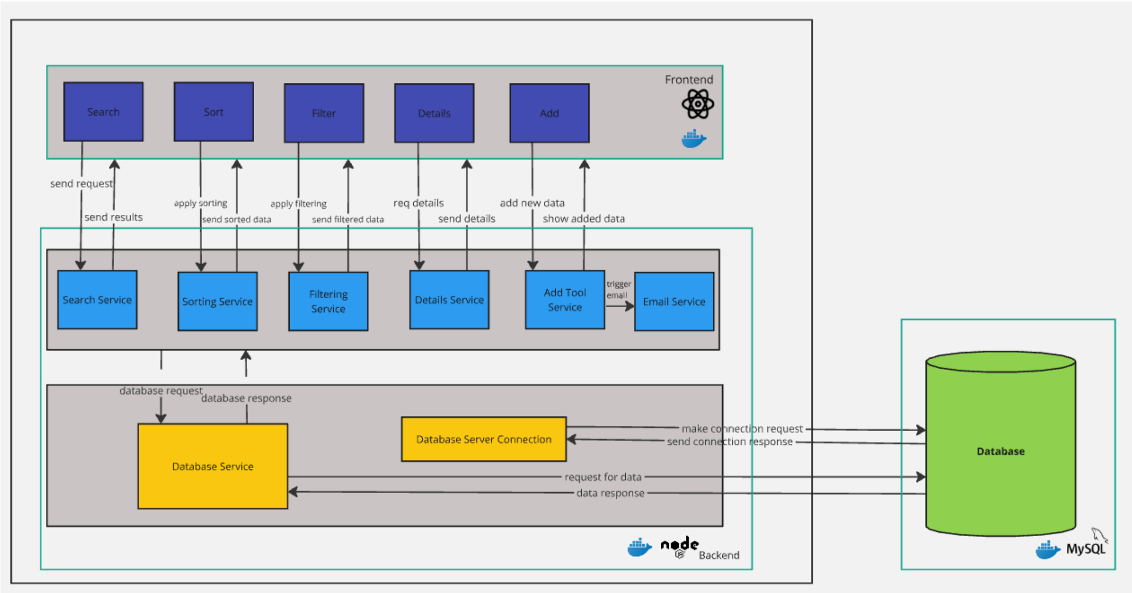

Figure 4.1: Architecture of the Platform

Figure 4.1 shows an architecture diagram in which a flow between a frontend, backend services and a database with modularized approach. It represents how these different components are interacting with each other. On the dashboard, we have different components for searching, sorting, filtering, adding a new software and a card view. The primary responsibility of the frontend is to communicate with the backend services which are done in the backend layer.

The top layer in the backend is the service layer which is a mediator between the frontend and database. It is responsible for taking the requests from frontend, applying the business logic before forwarding the request to database layer. In database layer, we have database connection component which is responsible for creating the correction with the database while starting the server. The database service is a gateway to execute command on the database and handle the response data. Additionally, when a new software is adding into the system, it also triggers the email service to send an email to the admin. This split-architecture principle makes our project clear in separation of concern, scalability and modularity as each piece of your application has a different purpose, and on long term it makes it manageable.

Sequence Diagram

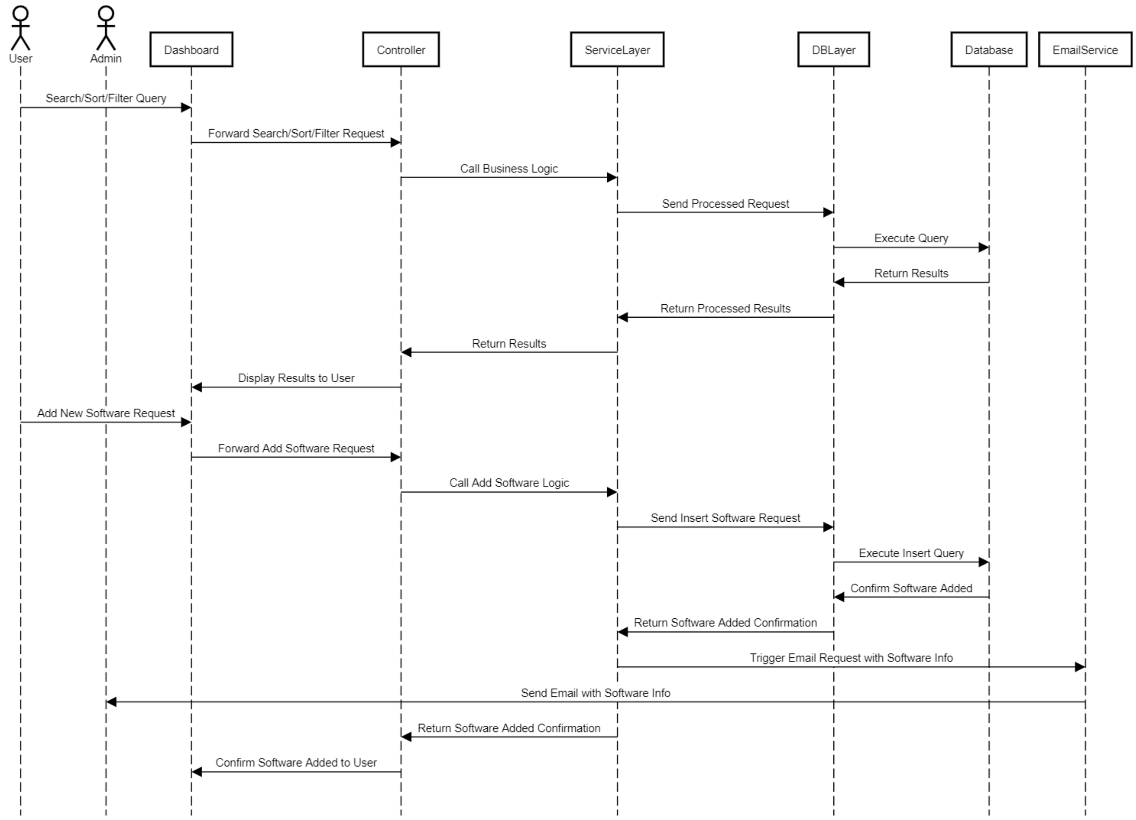

Figure 4.2: SequenceDiagram of the Platform

The sequence diagram is included to illustrate the flow of interactions between the different components of the platform during user operation. It helps visualize how the user, the UI, the service layer, and the database interact with each other in response to a specific action (such as searching or filtering AI planning tools). Figure 4.2 which shows a sequence diagram provides a clear understanding of the system's behaviour over time.

It has two actors: User and the admin.

The objects are: Dashboard, Controller, Service Layer, Database (DB) Layer, Database, Email Service.

We are showing two distinct interactions in the sequence diagram. The first interaction is about searing, sorting, filtering in the system. User creates a query from the dashboard that would make a request to the controller. Controller is responsible for frontend part and it sends a request to the service layer which manages the business logic for each type of request and then it is passed to the DB layer where SQL query is executed on the database. The database sends a response back to the layers one by one in reverse order and finally the result is displayed to the user.

The second one shows a sequence of adding a new software in the system and sending email notification to an admin about it. When the user submits a request for adding software, it comes from dashboard to frontend controller which checks for input validations as well and then it is forwarded to service layer where logic calculation happens for example – excellence score calculation, then the processed request is sent to DB layer where query is executed on the database. Once we have inserted the data into the database, backend triggers to send email for notifying the admin with software details and on the frontend user receives a confirmation of software addition to the platform.

4.4 Data Models

Main Data Model

The data model was created through a careful analysis of the key attributes required to comprehensively describe AI planning software. The parameters were chosen based on their relevance to the usability, and technical details needed by the platform's users, such as researchers, students, and AI practitioners. To decide which parameters to include, we conducted a review of existing platforms and identified the core metadata that users would need when searching, filtering, or comparing AI planning tools. Parameters like `implementationLanguage`, `operatingSystem`, and `sourceCode` were included to provide technical information essential for users or developers, while fields such as `year`, `contributor names`, and `plannerReferences` are of importance to researchers looking for academic information.

Our main model in the database has the following columns:

Column Name	**Explanation**
id	Unique identifier for each AI planning software
shortName	Short form of the AI Planning software
longName	Complete name of the AI planning software
contributors	Names of the individuals who contributed to the development or authors of the paper
year	The year when the AI planning software was made available
lastCommit	The date of the latest code modification on version control
explanationAbstract	A short description about the AI planning software
plannerReferences	Published paer link or related referenced used while development or research
documentation	Link of the documentation or a readme file
executable	Link of the executable file
sourceCode	Link of source code repository
implementationLanguage	Programming language(s) used to develop the AI planing software
operatingSystem	Supported operating system(s) of the AI planning software
environmentNotes	Dependencies or libraries which the tool needs to run on a system
planningLanguage	planning language or format the planning software uses
planningClasses	Classes of planning problems
planningType	Types of planning functionalities
insertedAt	Timestamp when the AI planning software was added to the database
excellenceScore	Score value indicating completeness

Table 4.1: Column names of table in database and its explanation

Column Name	Data Type	Example Value
id	varchar(36)	"4d8617b1-e4b1-423b-9300-eac485efd57f"
shortName	varchar(50)	"Blackbox"
longName	varchar(200)	"Blackbox"
contributors	varchar(255)	"Henry Kautz, Bart Selman"
year	int	1998
lastCommit	varchar(10)	"2016-06-20"
explanationAbstract	text	"Blackbox is a planning system that works by converting problems specified in STRIPS notation..."
plannerReferences	text	"https://www.cs.rochester.edu/u/kautz/satplan/blackbox/ "
documentation	varchar(500)	`https://gitlab.com/HenryKautz/BlackBox/-/blob/master/README.md?ref_type=heads`
executable	varchar(500)	"https://gitlab.com/HenryKautz/BlackBox"
sourceCode	varchar(500)	"https://gitlab.com/HenryKautz/BlackBox"
implementationLanguage	varchar(100)	"C"
operatingSystem	varchar(50)	"Linux, Windows, MacOS"
environmentNotes	text	"Note that compilation requires flex and bison"
planningLanguage	varchar(50)	"PDDL 1.2"
planningClasses	varchar(100)	"Classical"
planningType	varchar(100)	"Translation, Solving"
insertedAt	timestamp	"2024-09-10 00:06:30"
excellenceScore	float	8

Table 4.2: Column names, data types, and example values

Card View Model

The card view is an important UI element on the platform's homepage that provides users with a quick, compact view of each AI planning software via metadata. This view is important because it allows users to browse through a large collection of software efficiently, without being overwhelmed by all the available details. The card view only displays the most important attributes.The card view is considered separately because it acts as an entry point before getting into complete details. Once the user identifies the AI planning software then it can click on the card view to get detailed information.

On the homepage of the platform, there is a card view for each software and for which we are using the card-view model with only the important columns:

- shortName
- languages
- explanationAbstract
- year
- Contributors

- Excellence Score

4.5 Web Sketches

Before starting with the development, we first developed some web sketches which acts as a blue print for user interface of the platform. These web sketches are important as they bridge the gap between what had been written in requirements and its eventual implementation, showing the web design visually. With this visual model of the structure of the interface, we could clearly see where each feature fit and design it as per requirements.

Web Sketches for HomePage, DetailsPage, AddPAge, FilteringView and SortingView were made. There are listed in the Appendix section [A.1].

4.6 Score Calculation

We are adding another column to our database called `excellenceScore`, which tells the coverage of information about the AI planning software. Not all columns from the database are equally important. We assign a weight between 0 and 1 to each column based on its importance. ID and timestamp column has no relevance to this. The below table shows the columns with their individual weightage and its importance reason:

Column Name	Weight	Reason
shortName	0.5	Important for quick findings
longName	0.7	Importance based on complete name of the software
contributors	0.5	Importance regarding credibility
year	0.6	Importance as per published year
lastCommit	0.4	Importance based on recent activity
documentation	0.6	Importance based on documentation availability
explanationAbstract	0.8	Importannt as it gives summary of the planning software
plannerReferences	0.5	Importance regarding similar references
executable	0.5	Importance based on executable file availability
sourceCode	0.7	Importance based on source code availability
implementationLanguage	0.4	Important for implementation languages identification
operatingSystem	0.3	Important for identifying compatible operating systems
environmentNotes	0.2	Important for identifying dependencies and libraries
planningLanguage	0.5	Important for planning languages identification
planningClasses	0.4	Importannt for planning class identification
planningType	0.4	Importannt for planning functionality type identification

Table 4.3: Table showing column names, their weights, and reasons for the importance of each.

This is the formula to calculate Excellence Score:

$$\begin{aligned}\text{Excellence Score} = {} & (0.5 \times \text{Short Name}) + (0.7 \times \text{Long Name}) + \\ & (0.5 \times \text{Contributors}) + (0.6 \times \text{Year}) + \\ & (0.4 \times \text{Last Commit}) + (0.6 \times \text{Documentation}) + \\ & (0.8 \times \text{Explanation Abstract}) + (0.5 \times \text{Planner References}) + \\ & (0.5 \times \text{Executable}) + (0.7 \times \text{Source Code}) + \\ & (0.4 \times \text{Implementation Languages}) + (0.3 \times \text{Operating Systems}) + \\ & (0.2 \times \text{Environment Notes}) + (0.5 \times \text{Planning Language}) + \\ & (0.4 \times \text{Planning Classes}) + (0.4 \times \text{Planning Type})\end{aligned}$$

The Excellence Score value can range from a minimum of 0.5 to maximum of 8.0. The minimum value is 0.5 as either of shoftName or longName is required. If the row has all the column information, the weights adds to 8.0.

We categorize them based on the Excellence Score value in the frontend as follows:

Star Number	**Excellence Score Range**
5 Star	7.0 and above
4 Star	Less than 7.0 and more than 5.5
3 Star	Less than 5.5 and more than 4.0
2 Star	Less than 4.0 and more than 2.5
1 Star	Less than 2.5

Table 4.4: Excellence Score Categories of AI planning software

5 Implementation

We discuss the technical aspects concerning our solution. We find the best technology stack for our platform development, delve into data collection part and then later discussing the implementation details showing modular structure.

5.1 Technical Details

5.1.1 Frontend

We are using ReactJS which is a popular JavaScript library for frontend development. React plays an important role due to its lightweight nature and it is useful in creating dynamic and responsive web pages [36]. The frontend of the platform has features like search, filtering and sorting of AI planning software which can be easily handled through React's component-based architecture. It also allows the systems to remain scalable and flexible over time as features are scaled or modified over time. This makes React as a best choice for our platform development.

5.1.2 Backend

We are using Node.js for our backend development. Node.js is JavaScript runtime environment which is used to create servers [37]. In this project, Node. It is essential for working with backend operations such as Application Programming Interface (API) requests and data processing. Since Node.js is built to organize event-based operations in a asynchronous and non-blocking way, it can serve lots of user requests concurrently with continued responsiveness on heavy input and output operations. In our implementation, it is quick to make API requests with Node.js to fetch data on request. Node.js acts as a mediator between frontend and database requests.

5.1.3 Database

We are using MySQL as database for our project development. MySQL is a common, open and well-supported relational database management system (DBMS) which is used to handle storage and management of AI planning tool metadata. MySQL is important with this project as it provides data integrity and scalability [38]. If a platform provides SQL queries, it is able to filter and sort massive sets of data, fetching only the relevant rows that are requested by users. It also has the ability to scale and handle large datasets. The metadata about AI planning software is structured into rows and columns and MySQL is convenient for this purpose.

5.1.4 Containerization

Docker is being used to containerize each component of the platform i.e frontend (React), backend (Node.js), and the MySQL database. It makes sure that the platform is consistent and portable across various systems. Docker allows the entire platform and its dependencies to be packaged so it is straightforward to deploy and manage [39]. By isolating every single component in separate containers, it allows to do independent updates to each of them if required. Additionally, this containerization approach also makes it easier to onboard new developers and collaborators because they can run the platform in a reproducible environment easily. This also supports our key aspect of improvised collaboration for developers.

5.2 Data Collection

Raw data of the AI planning software was received from my supervisor as an Excel sheet. There were hundreds of entries to the dataset on AI planning tools, but it needed a lot of preprocessing to ensure accuracy and completeness before the data could be integrated into the platform.

There were formatting inconsistencies in the text e.g. differing symbols/characters, missing or incorrect information from some fields, and incomplete records. Every row of the dataset needed to be inspected and validated as accurate. This included cross verifying the entries by manually, correcting them whenever required we also filled in missing data and validated all these visiting the reference links provided.

In preprocessing phase, we corrected symbols which were mostly found in contributor names. New columns were also introduced such as Planning functionality type where we filled in data to complete all rows and columns as per the requirements of the platform. Row-by-row cross-verification was done, mistakes were corrected and the information was updated.

This preprocessing step had to be done in order to make the data clean, accurate and ready for adding it into our platform.

5.3 Implementation

Step 1: Database Setup (MySQL):
The system environment is set up by installing MySQL installer for Windows (8.4.0) and proceeded with default installation and configuration steps. Along with this, MySQL Workbench is installed and configured, which serves as a Graphical User Interface (GUI) for managing the database. The `ai_planning_software` table is created in the database following the model discussed above. The refined Excel sheet was imported into the MySQL database using a Python script.

Each row in the database is uniquely indentified by an UUID(Universally Unique Identifier). An `inserted_At` column is introduced in the database where each row is assigned a timestamp with a delay of 5 seconds. Lastly, the `excellenceScore` column is also added to the table, and based on our excellence score formula (discussed earlier), values are added to each individual row. By doing so, our database setup is complete and ready to use.

Step 2: Backend Development (Node.js):
The system environment is set up by installing Node.js (v18.13.0) from the official website. The Express framework is installed, which is required for backend operations such as handling requests to the database.

By following the principles of layered architecture, a modular project structure for the backend is created, which we explain as follows: The starting point of the application is `app.js` where the requests from the frontend are received. It, in turn, calls the routing file to handle the incoming requests and maps them to backend API calls. The service layer acts as an intermediary between the routing and database layers. It contains the business logic and the pre-processing logic that does not deal with the database directly. In the service layer, we have written the business logic for searching, sorting, and filtering functionality. Next, the service layer request moves to the models file, which is responsible for interacting with the database and executing SQL commands. The SQL commands for fetching the AI planning software and inserting new software information are executed from this file. The data request and response happens via API endpoints and it follows standard data format such as JSON for response data.

We have a configuration file containing credentials for handling the database connection. Lastly, we have a utility file responsible for sending emails to the admin after adding software to the platform.

Step 3: Frontend Development (React):
We set up the frontend environment by installing React and Axios for making API requests from the frontend. Here, we also follow a modular approach for creating the frontend for the sake of reusability and maintainability. The `index.js` is the entry point of the frontend application, which contains the component to render at first. In our project, this is the `App.js` component.

We have a `components` folder where we have created different component files as per the functionality. The different functionalities and views include Card View, Details View, Filtering Sidebar, Adding a Software, Pagination, About Us, Privacy Policy—each of which has a different component file. We also have separate component files for the Header, Footer, and Spinner, which are reusable across various pages. A custom hook handles asynchronous data fetching, and the components are used to display data and allow user interaction.

Additionally, we are performing validations for the Add functionality, ensuring that the data being inserted is of a valid type. For example, email validation and year validation.

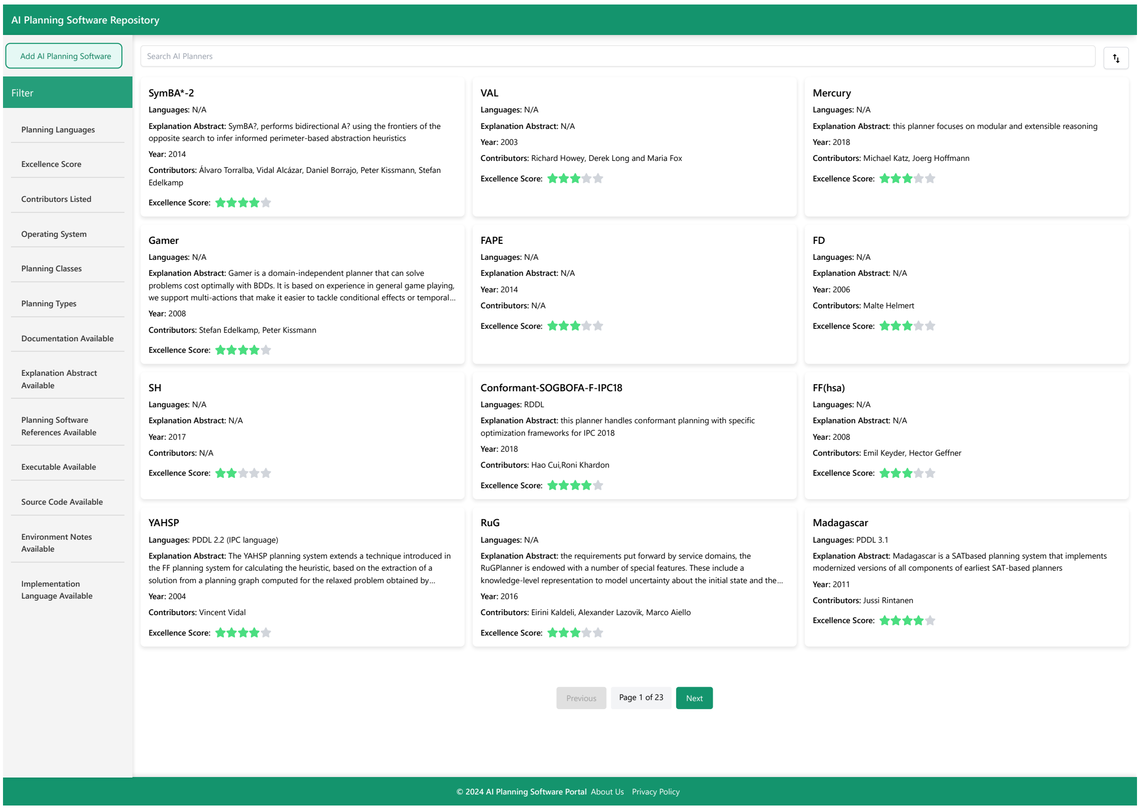

Figure 5.1: Implemented Interface

Step 4: Integration with Docker for Containerization:
We have separate Dockerfiles for the frontend and backend, which contain commands to install the necessary dependencies and libraries for the respective projects. Three individual containers are created for the frontend, backend, and database.

We have a `docker-compose.yml` file that ensures the running of multiple containers, and it also ensures that different containers can communicate with each other. This approach makes the project system-independent and ensures that all installations have the same configurations, making it easy to reproduce.

Step 5: Testing and Debugging:
Test cases are written for both the frontend and backend to check the reliability and functionality of the platform. In the frontend, we have test cases for header and footer rendering, validation of input fields, adding functionality, and filter drawer functionality. The tests are written using Jest and React Testing Library.

6 Evaluation

We analyze and evaluate our solution by different methods. We present the qualitative and quantitative user study and also we evaluate technical performance of the platform. Lastly, we reflect the discussion based on the results and highlight the improvements made after this study.

6.1 User Study

6.1.1 Qualitative User Study

Qualitative user studies target the reasons for specific user behaviors by working with data which are non-numbered. These studies an aspect of user needs and motivations in interviews, observations, some open-ended questionnaires. It helps in understanding user personal experience and preferences [40].

We performed the study with 6 users of three different academic levels (Bachelors, Masters and PhD students) in order to conduct a standard evaluation for the AI planning software portal. Every user worked with the platform and shared feedback on the entire User Interface (UI), how usable it is, what they like, what they dislike and also improvements. That feedback tells us more about how to develop and deliver best features to improve the experience with the platform.

Methodology

The questions asked to the users were:

1. How do you find the user interface (UI)?
2. Was it difficult for you to understand the UI?
3. What did you like or dislike about the UI?
4. What improvements would you suggest?

The raw answers from the users are in the table in Appendix [A.2.1]

6.1.2 Quantitative User Study

Quantitative user studies brings measurable data together and analyze this data to recognize specific changes in individual behaviours or populations. Quantitative studies provide an objective view by allowing the aspects like task completion time, success rate or error rate to be quantified which helps in identifying those areas that needs to be improved. This is also an important part of user study [41].

Users were asked how they would navigate the platform and complete the tasks. The tasks were as follows:

1. Find AI planning software related to scheduling
2. Sort AI planning software by published year
3. Filter AI planning software by excellence score
4. Add a new software into the platform

Following metrics were evaluation from this study [41]:

- **Time on Task**: It measures how long it takes a user to perform a task. This helps us understand if the user completes the task within a threshold or takes more time.
- **Success Rate**: The percentage of tasks successfully completed by a user. This is a key measure to determine whether users are able to complete the task or not.
- **Error Rate**: The number of mistakes a user makes while performing a task. This metric helps identify usability issues.

6.2 Technical Evaluation

Technical evaluation of **page load time** is an extremely important element to take into account as it substantially influence user experience and overall performances on a platform. It is an important metric that measures how quickly a web page fully loads and becomes usable for the user.

We also have done feature-level performance evaluation by calculating the response time for searching, sorting and filtering functionality.

These metrics were checked on Google Chrome Browser(v129.0.6668.71) via developer tools.

6.3 Results

In this section, we see the results from the study conducted as mentioned above.

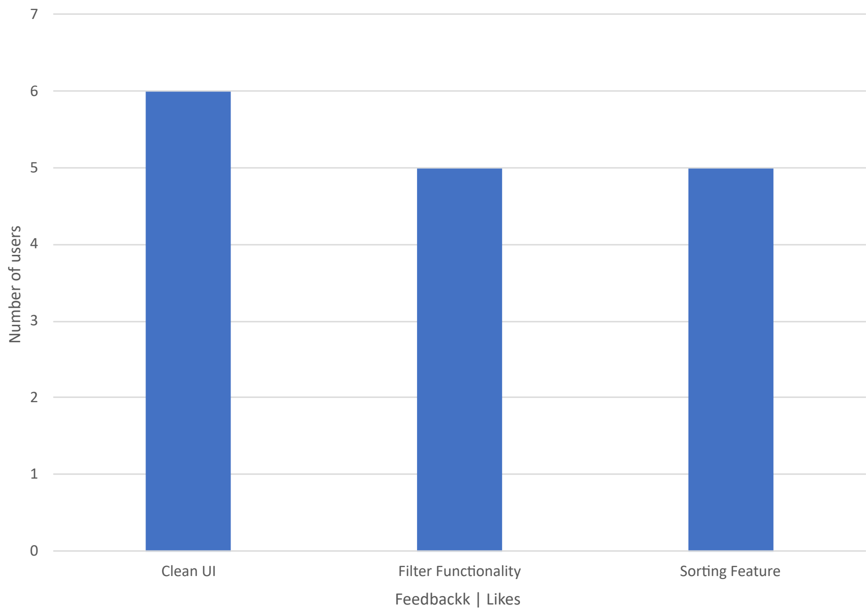

Figure 6.1: User Feedback about suggestions

6.3.1 Conclusion from Quantitative Study

- **UI:** All users were happy with the user-friendly UI, as they felt it was easy to use and clutter-free.
- **Sorting and Filtering:** Users were able to perform the operations effortlessly.
- **Add Button:** 3 users felt that the "Add AI Planning Software" button was not obvious and could easily be missed as it was located below all the filters.
- **Excellence Score:** Two of six participants, one reported finding the star rating unclear. This user wanted further clarification or a redesign of this feature.

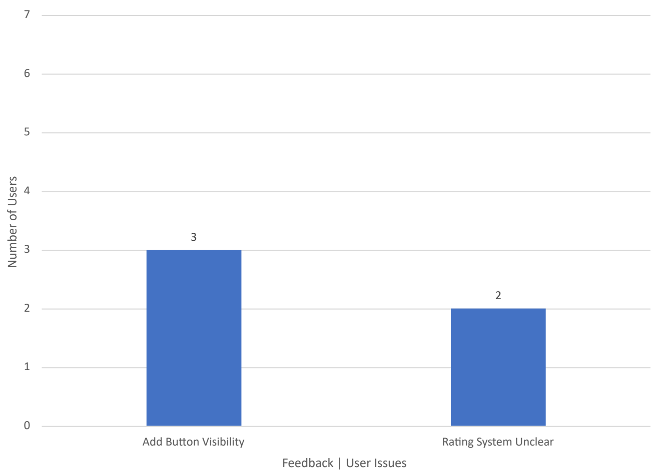

Figure 6.2: User Feedback about suggestions

6.3.2 Conclusion from Qualitative Study

From User Time Task study, we can see how much time each user took for completing the tasks. The first three tasks involving searching, sorting and filtering should be quick and we find that users could complete the task within 20 seconds. The last task was about adding a new software to the platform which is suppossed to take a little more time than the first three tasks as it involves filling up the input fields for a new software and then submitting the data. For task four, average time taken by users was 192 seconds. (given that they had all software related information with them) Overall, all the tasks are completed quicky and we can say, there was no UI confusion on the website.

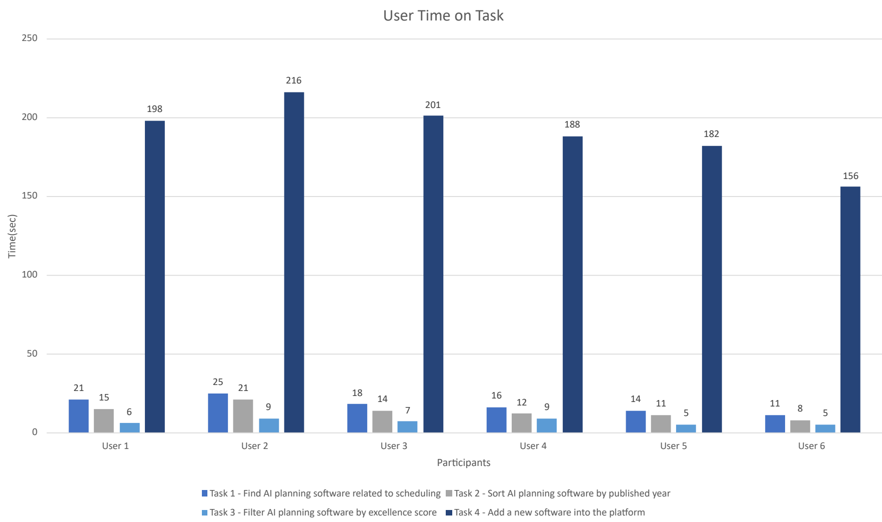

Figure 6.3: User Time Task

While performing the tasks, it looks like most users were able to complete the task successfully and with no errors. However, 3 users could not successfully complete the task 1 of finding a software related to scheduling. They located a diffrent software. The reason for this unsucessful task was not completely from the user side but rather a system error which returned few addtional unrealted software. It was found that search was checking url fields from the database such as documentation, source code, executable links and which let to additional results. We talk about next step in the discussion part later in the chapter.

And 2 users could not locate the Add AI planning software button in the UI as they naviagted to different pages during the time completion which is counted into an error. If the user is deviated or misguided on the UI, that is considered as an error.

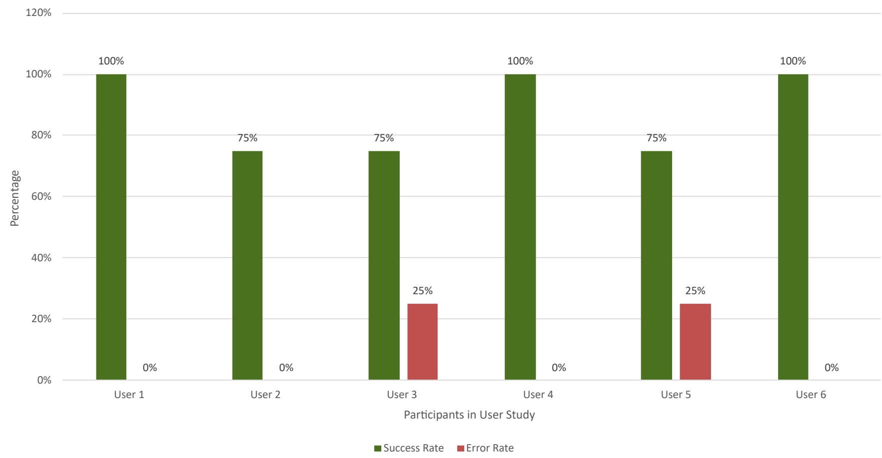

Figure 6.4: Success and Error rate of User Tasks

6.3.3 Conclusion from Technical Evaluation

As per the Google's Core Web Vitals, there are categories of Good, Moderate and Poor page load time. If it is below 2.5s, then it is considered good, between 2.5s and 4s is moderate and above 4s is a poor page load time [42].

Our platform has a PageLoadTime of 113ms which is excellent.

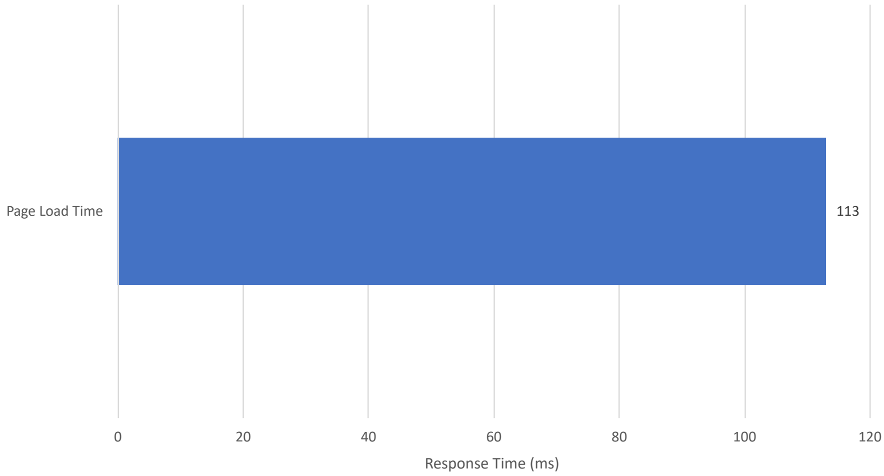

Figure 6.5: Performance Evaluation

In the feature-level performance evaluation, different set of inputs for search, different options of sorting and filtering were checked for response time. It was found that on average, Searching took 13.9ms, Filtering took 11.6ms and Sorting took 21.46ms to complete.

This result also shows excellent feature-level response time.

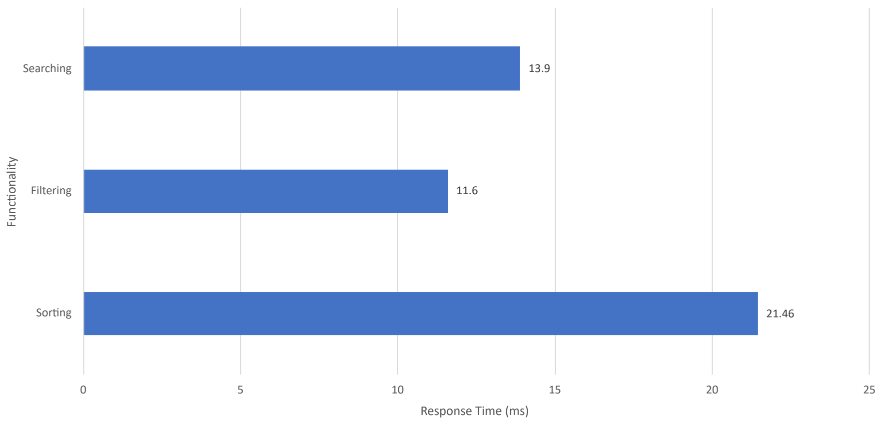

Figure 6.6: Feature level Performance

6.4 Discussion

Common strengths of the platform identified from the qualitative study were a **clean UI and good organization**. While these strengths make the platform easier to adopt for a wider audience, improvements could be made by making the button more visible, clarifying the excellence score, and improving the search functionality.

Using the feedback from the study, we **improved our platform** by

- Re-positioning the add button to the top
- Applying a uniform color to the excellence score, and adding a tooltip
- Modifying the search logic so that it excludes all columns which contain URLs

Overall, user study was importance for the platform and by incorporating these changes, the overall user experience is improved, making the platform more user-friendly for new users.

7 Conclusions and Outlook

In the last chapter, we summarize our thesis research and our solution. We present how well we have addressed our research question. We delve into its limitations and later, we outline on how this research affects the field of AI planning.

7.1 Summary

The main outcome from this thesis research is the development of a platform that solves the problem of unavailability of a unified platform in AI planning domain. At first, we researched about current AI planning domain and we find that there had been advancements going on in the field but the information is scattered at multiple websites, repositories, or conference papers. We also did research about the available platforms in related fields such as ML and AI in medical industry. The findings showed that such a platform is still not available for AI planning domain.

We develop and design the platform that brings all the scattered information about AI planning software to one single place. This helps users to discover, compare different AI planning software as per their requirements. The platform is designed and developed by considering different user groups such as students, researchers, practitioners, and industry specialists from the domain. We have made the platform user-friendly to cater different user groups and also for widespread adoption of the platform.

The research identified several key elements crucial for designing and developing a platform that fosters widespread adoption of AI planning software. The platform's user-centric design with features like search, filter, and sort proved important for meeting the needs of different user groups, including students, researchers, and AI practitioners. Another finding was about organising the meta-data in a format so that it is uniform across different AI planning software. The platform's architecture was designed to be modular which ensures scalability, maintainability and fosters adoption of the system.

7.2 Limitations

Although the platform was implemented successfully, there were limitations throughout the platform's We delve into its development process. The availability and quality of the data was one of the limitations. The data loaded into the platform needed a level of cleaning and preprocessing which was done manually. It needed browsing through various sources with inconsistent meta-data. A bunch of AI planning software had almost all the required information available but there were a few with limited or no information at all.

The platform also has the feature of manually adding the AI planning software to the platform. While the platform makes it easy to add new AI planning software for further discover on our platform, the integration still relies on a manual submission from users from the domain. Manual data entry always has a risk of making mistakes while writing. Also automating the identification of new AI planning software from external sources may additionally make the coverage of the AI planning software up-to date.

We did a user study with different user groups, student in their bachelors, masters and PhD program and received feedback about the platform. This helped in gaining insights from the user point of view. However, making the platform available to even wider audience publicly would give us even more feedback about its adoption.

7.3 Outlook

This platform presents several opportunities for future improvements and broader impacts on the AI planning field. The current system depends on manual entry of the AI planning software from research papers, repositories, scientific publications, automation is a possibility which gives users with up-to date software information. Such functionality would enhance the platform.

Apart from enhancements, the platform has capacity to influence other research directions. By providing centralized, easy access to AI planning software, the platform can facilitate the study of AI planning systems, encouraging further research into comparisons of existing software. Researchers could also study trends in the evolution of AI planning algorithms by using the platform's uniform and standardized metadata. The platform could also be used in an educational environment as a resource for teaching AI planning topics and enabling students to explore and compare AI planning software helping in understanding artifacts of individual software without going through numerous websites. It also gives an opportunity to create case studies that focus on software selection and analysis based on the attributes. The platform's modular design can serve as a model for creating similar repositories in other AI subfields such as natural language processing, encouraging collaboration and knowledge sharing.

Bibliography

[1] I. Georgievski. “Conceptualising Software Development Lifecycle for Engineering AI Planning Systems”. In: *AI Engineering Journal* (2023) (cit. on p. 13).

[2] J. Hendler, A. Tate, M. Drummond. “AI Planning: Systems and Techniques”. In: *AI Magazine* 11.2 (1990), pp. 61–77 (cit. on pp. 13, 17).

[3] E. Alnazer, I. Georgievski, N. Prakash, M. Aiello. “A role for HTN planning in increasing trust in autonomous driving”. In: *2022 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2022, pp. 1–7 (cit. on p. 13).

[4] I. Georgievski, F. Nizamic, A. Lazovik, M. Aiello. “Cloud ready applications composed via HTN planning”. In: *2017 IEEE 10th Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE, 2017, pp. 81–89 (cit. on p. 13).

[5] I. Georgievski, T. A. Nguyen, F. Nizamic, B. Setz, A. Lazovik, M. Aiello. “Planning meets activity recognition: Service coordination for intelligent buildings”. In: *Pervasive and Mobile Computing* 38 (2017), pp. 110–139 (cit. on p. 13).

[6] I. Georgievski, M. Z. Shahid, M. Aiello. “AI temporal planning for energy smart buildings”. In: *Energy Informatics* 6.Suppl 1 (2023), p. 18 (cit. on p. 13).

[7] K. Bradbrook, G. Winstanley, D. Glasspool, J. Fox, R. Griffiths. “AI Planning Technology as a Component of Computerised Clinical Practice Guidelines”. In: *Artificial Intelligence in Medicine. AIME 2005*. Ed. by S. Miksch, J. Hunter, E. T. Keravnou. Vol. 3581. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 38–47. DOI: 10.1007/11527770_26 (cit. on p. 13).

[8] M. Camilli, M. Felderer. “Towards Risk Modeling for Collaborative AI”. In: *Collaborative AI Journal* (2021) (cit. on p. 13).

[9] M. Roberts, A. Howe. “A Call for Participation: Supporting Large Scale Studies of Planning Systems”. In: *Proceedings of the ICAPS 2007 Workshop on the International Planning Competition: Past, Present and Future*. 2007 (cit. on p. 13).

[10] I. Georgievski. “Engineering AI Planning Systems”. See Barrier 7 for details on challenges in AI planning systems. Kumulative Habilitationsschrift. Stuttgart, Germany: Universität Stuttgart, Apr. 2024 (cit. on p. 13).

[11] I.-L. Yen, F. Bastani, F. Mohamed, H. Ma, J. Linn. “Application of AI planning techniques to automated code synthesis and testing”. In: *14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings.* 2002, pp. 131–137. DOI: 10.1109/TAI.2002.1180797 (cit. on p. 13).

[12] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. Sri, A. Barrett, D. Christianson, et al. “Pddl| the planning domain definition language”. In: *Technical Report, Tech. Rep.* (1998) (cit. on p. 14).

[13] I. Georgievski. "Towards engineering AI planning functionalities as services". In: *International Conference on Service-Oriented Computing*. Cham: Springer Nature Switzerland, 2022, pp. 225–236. DOI: 10.1007/978-3-030-97509-6_16 (cit. on p. 14).

[14] I. Georgievski, U. Breitenbücher. "A vision for composing, integrating, and deploying AI planning functionalities". In: *2021 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2021, pp. 166–171. DOI: 10.1109/SOSE52839.2021.00029 (cit. on p. 14).

[15] M. Ghallab, D. Nau, P. Traverso. *Automated Planning: Theory and Practice*. San Francisco, CA: Elsevier, 2004. ISBN: 978-1-55860-856-6 (cit. on p. 14).

[16] I. Georgievski. "Towards Engineering AI Planning Functionalities as Services". In: *Journal of AI Engineering Services* (2022) (cit. on pp. 14, 18).

[17] M. Litoiu, C. Barna. "A performance evaluation framework for Web applications". In: *Computer Science and Engineering, York University* (July 2012) (cit. on p. 14).

[18] Z. Tu. "Research on the Application of Layered Architecture in Computer Software Development". In: *Journal of Computing and Electronic Information Management* 11.3 (Nov. 2023), pp. 34–38. DOI: 10.54097/jceim.v11i3.08. URL: https://drpress.org/ojs/index.php/jceim/article/view/14398 (cit. on p. 15).

[19] W. M. C. J. T. Kithulwatta, W. U. Wickramaarachchi, K. P. N. Jayasena, B. T. G. S. Kumara, R. M. K. T. Rathnayaka. "Adoption of Docker Containers as an Infrastructure for Deploying Software Applications: A Review". In: *Advances on Smart and Soft Computing*. Ed. by F. Saeed, T. Al-Hadhrami, E. Mohammed, M. Al-Sarem. Singapore: Springer Singapore, 2022, pp. 247–259. ISBN: 978-981-16-5559-3 (cit. on p. 15).

[20] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. '. Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3.1 (2016). Box 2 | The FAIR Guiding Principles, p. 160018. DOI: 10.1038/sdata.2016.18. URL: https://doi.org/10.1038/sdata.2016.18 (cit. on pp. 15, 27).

[21] I. Georgievski, U. Breitenbucher. "A Vision for Composing, Integrating, and Deploying AI Planning Functionalities". In: *Journal of AI Planning Vision* (2021) (cit. on p. 17).

[22] N. Widulle, L. M. V. da Silva. "Investigating the Use of AI Planning Methods in Real-World CPS Use Cases". In: *Journal of Cyber-Physical Systems* (2022) (cit. on p. 17).

[23] F. Mohr. "Issues of Automated Software Composition in AI Planning". In: *Department of Computer Science, University of Paderborn* (2024). felix.mohr@upb.de (cit. on p. 17).

[24] C. Domshlak. *Slides: Automated (AI) Planning, Czech Technical University, Prague*. 2022 (cit. on pp. 17, 18).

[25] R. Sabbadin, F. Teichteil-Königsbuch, V. Vidal. "Planning in Artificial Intelligence". In: *A Guided Tour of Artificial Intelligence Research: Volume II: AI Algorithms*. Ed. by P. Marquis, O. Papini, H. Prade. Cham: Springer International Publishing, 2020, pp. 285–312. ISBN: 978-3-030-06167-8. DOI: 10.1007/978-3-030-06167-8_10. URL: https://doi.org/10.1007/978-3-030-06167-8_10 (cit. on p. 17).

[26] E. Kaldeli, A. Lazovik, M. Aiello. "Domain-independent planning for services in uncertain and dynamic environments". In: *Artificial Intelligence* 236.7 (2016), pp. 30–64 (cit. on p. 17).

[27] M. Georgeff. "Communication and Interaction in Multi-Agent Planning Systems". In: *Proceedings of the Third National Conference on Artificial Intelligence*. Menlo Park, Calif.: American Association for Artificial Intelligence, 1982 (cit. on p. 18).

[28] A. Green, B. J. Reji. *Planning.Wiki - The AI Planning & PDDL Wiki*. Accessed: 2023-09-24. n.d. URL: https://planning.wiki/ (cit. on pp. 21, 26).

[29] J. Hoffmann. *The FF Planning System*. https://fai.cs.uni-saarland.de/hoffmann/ff.html. Accessed: October 7, 2024 (cit. on p. 21).

[30] *Fast Downward*. https://www.fast-downward.org/. Accessed: October 7, 2024 (cit. on p. 21).

[31] *International Planning Competitions (IPC)*. https://www.icaps-conference.org/competitions/. Accessed: October 7, 2024 (cit. on pp. 21, 26).

[32] *International Conference on Automated Planning and Scheduling (ICAPS)*. https://www.icaps-conference.org/. Accessed: October 7, 2024 (cit. on p. 21).

[33] *GitHub*. https://github.com/. Accessed: October 7, 2024 (cit. on pp. 21, 26).

[34] J. N. van Rijn, B. Bischl. "OpenML: A Collaborative Science Platform". In: *Journal of Machine Learning* (2013) (cit. on pp. 22, 23, 26).

[35] AIMe. *AIMe: AI in Medicine*. Accessed: 2023-09-24. n.d. URL: https://aime-registry.org/about (cit. on pp. 22, 23, 26).

[36] *Introducing React Dev*. Accessed: 2024-10-04. 2023. URL: https://react.dev/blog/2023/03/16/introducing-react-dev (cit. on p. 37).

[37] *Node.js Official Website*. Accessed: 2024-10-04. n.d. URL: https://nodejs.org/en (cit. on p. 37).

[38] *MySQL 8.0 Reference Manual*. Accessed: 2024-10-04. n.d. URL: https://dev.mysql.com/doc/ (cit. on p. 37).

[39] *Docker Official Documentation*. Accessed: 2024-10-04. n.d. URL: https://docs.docker.com/ (cit. on p. 38).

[40] L. Banwell, G. Coulson. *Users and user study methodology: the JUBILEE project*. https://informationr.net/ir/9-2/paper167.html. Accessed: 2024-10-04. 2004 (cit. on p. 41).

[41] *Usability Testing Metrics*. Accessed: 2024-10-04. n.d. URL: https://www.uxtweak.com/usability-testing/metrics/ (cit. on p. 42).

[42] *What is Page Load Time?* Accessed: 2024-10-04. n.d. URL: https://www.debugbear.com/docs/page-load-time (cit. on p. 45).

All links were last followed on October 04, 2024.

A Appendix

A.1 Web Sketches

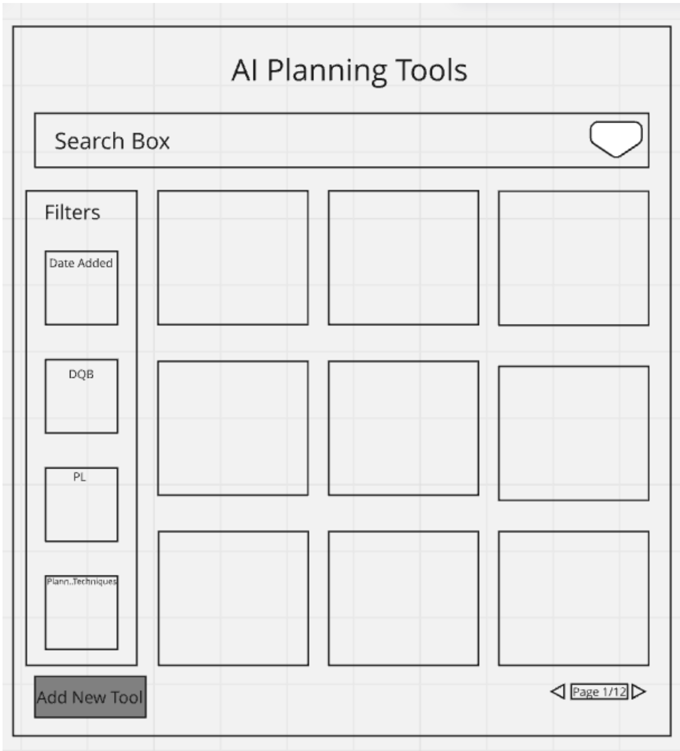

Figure A.1: HomePage

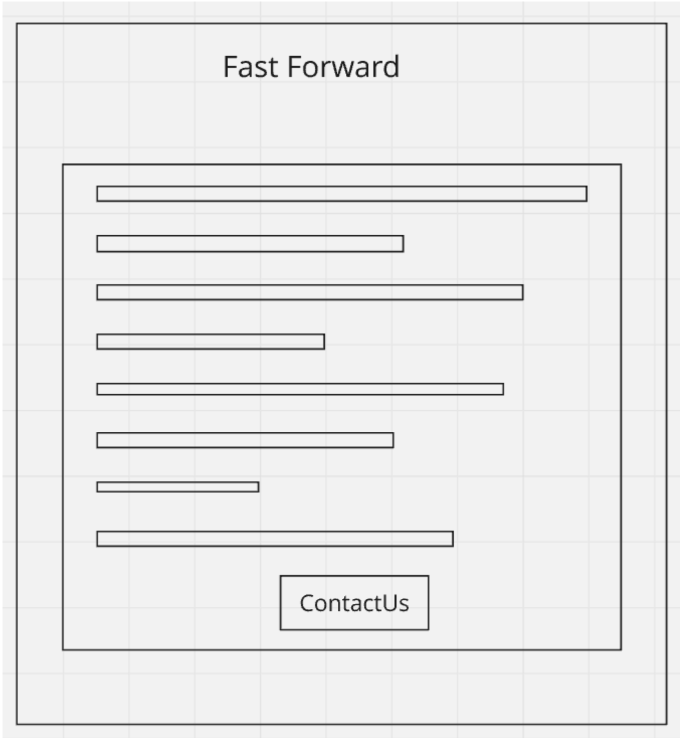

Figure A.2: Details Page

Figure A.3: Add Page

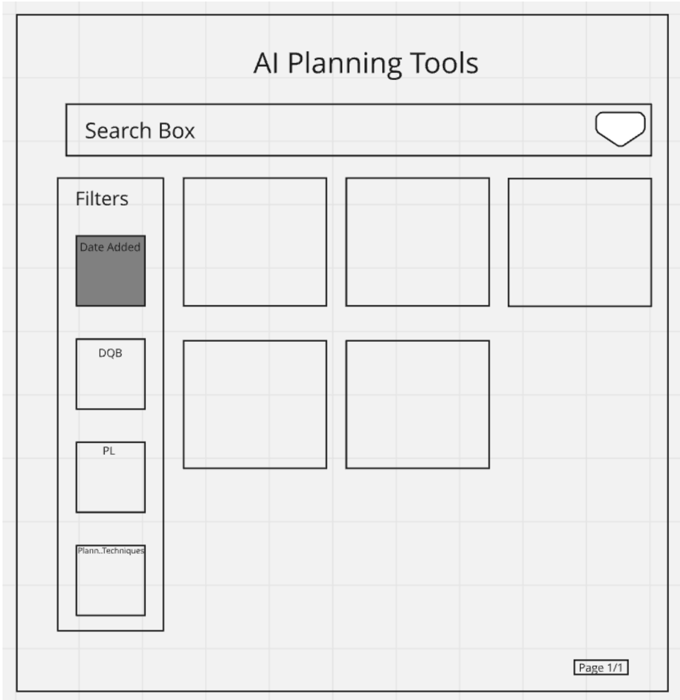

Figure A.4: Filtering View

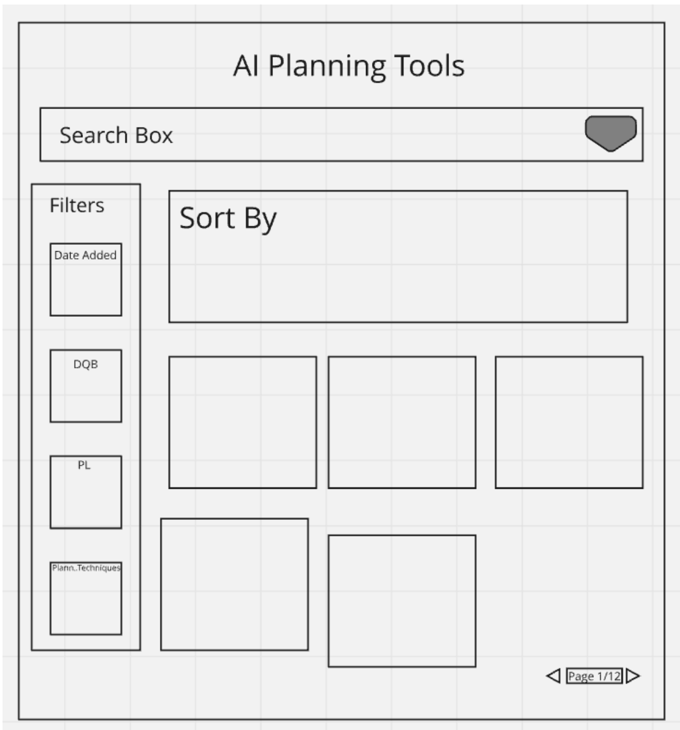

Figure A.5: Sorting View

A.2 User Study

A.2.1 Qualitative

User	**UI Clarity**	**Ease of Use**	**Likes**	**Dislikes**	**Suggestions**
User 1 (Student / Chemistry)	✓ Clean UI	✓ Easy	Clean and nice	Add button hidden	Make add button more visible
User 2 (Student / Bachelor)	✓ Good	✓ Easy	Filter functionality	Too many cards	Limit cards to 6 per page
User 3 (Student / Bachelor)	✎ Star rating unclear	✓ Easy	Sorting feature	Star rating system unclear	Reposition add tool button
User 4 (Student / Master Electrical)	✓ Good and compact	✓ Easy	Pagination is okay	None	Improve pagination layout
User 5 (PhD Student)	✓ Compact, clutter-free	✓ Easy	Clarity compared to OpenML	None	None
User 6 (Student /Master)	✓ Intuitive layout	✓ Easy	Card View	-	None

Table A.1: User feedback on UI Clarity, Ease of Use, Likes, Dislikes, and Suggestions.

A.2.2 Quantitative

User	Tasks Completed	Total Tasks	Errors	Comments
User 1	4	4	0	No errors
User 2	3	4	0	Task 1 failed
User 3	3	4	1	Task 1 failed, Error during Task 4
User 4	4	4	0	No errors
User 5	3	4	1	Task 1 failed, Error during Task 4
User 6	4	4	0	No errors

Table A.2: Error rate, task completion, and user comments from qualitative user study