## 1 Introduction

Online shopping platforms like Amazon and eBay have become the go-to sources for a vast range of products. However, users often face difficulties when searching for specific items, especially products with more complex or technical specifications, such as electronic devices, computer components, or specialized tools. These challenges can lead to frustration and even cause users to switch platforms in search of a better experience. The root of these difficulties lies in the individuality of search behavior, which varies between users. Some rely on generic search terms, such as "laptop," while others prefer highly specific queries, like "Dell XPS 15 9530 with Intel Core i7 and 32GB RAM." Our study found that search behavior is roughly evenly divided between these two approaches, making it essential for e-commerce platforms to optimize their search functionalities to accommodate diverse user needs. At the same time, this presents a challenge, as creating a platform that satisfies all users equally is difficult. In addition to search queries, filtering options play a crucial role in the online shopping experience. Filters help users refine their results based on criteria such as price, size, brand, technical specifications, or customer ratings. However, ineffective or overly complex filtering systems can hinder rather than enhance the search process. If filters do not align with user expectations, are difficult to use, or fail to narrow down results efficiently, users may become frustrated and abandon their search. Therefore, improving search functionalities is essential to enhance user experience, increase customer satisfaction, and ensure platform loyalty.Building upon these challenges, we propose an alternative approach to traditional search mechanisms. Instead of relying on predefined SQL templates, we suggest dynamically generating SQL queries using machine learning techniques. By leveraging AI to generate queries in real time, the system can automatically interpret the user's intent based on their input, providing more precise and relevant search results initially. If the AI performs at a high level, this approach could eventually eliminate the need for manual filtering mechanisms, as the AI would inherently incorporate these preferences into the generated query. This shift could significantly enhance human-computer interaction, making the search process more intuitive, efficient, and user-centric. To determine whether such a scenario is truly possible and to understand how users would interact with this approach, we conducted a study. To carry out this study, we implemented a prototype interface to evaluate user behavior and gather insights on the effectiveness of dynamically generated queries in direct comparison to traditional static templates.

### 1.1 old

In order for a user to interact with a website, an interactive element such as an input text is typically provided. Many of these input text-elements work by having a predefined SQL-Query-Template, for which the users input gets concatenated. This query then gets passed to the Database-Management-System (DBMS), which executes it and returns the result. Although this approach seems simple at first, it has some pitfalls which need to be addressed, as it is prone to attacks like SQL-Injections. According to the Open Worldwide Application Security Project (OWASP), which is a non-profit organization that aims to improve the security of software projects, SQL-Injections are still in the Top 10 Security Risks of 2021 and it is likely that they will remain in the Top 10 of 2025, taking into account that they were the number one attack in the Top 10 of 2017 [**?** ].

### 1.2 Generating SQL-Queries dynamically

Instead of relying on predefined SQL-Templates, we would like to propose the idea of generating SQL-Queries dynamically using Machine-Learning-Techniques.
By generating SQL-Queries dynamically, the users wishes provided in the input text can be taken into account by the

AI automatically, providing more specific search results initially. Given that the AI performs well enough, this could lead to future websites dropping filtering mechanisms entirely, such as filtering by specific brands or other features, as the AI would reflect the users wishes automatically in the generated Query, further enhancing the Human-Computer Interaction. This new kind of interaction is highlighted in Fig. 1.

### 1.3 Why a privacy-preserving AI is necessary

Although the idea of generating SQL-Queries automatically sounds intriguing, there are some practical challenges that have to be taken into account, when deploying the AI into a software project. Under the assumption that the AI performs well and that it is unrestrained, nothing would prevent the user from requesting search results from the AI, that contain confidential information like passwords, email-addresses or other criteria, which poses a similar threat that are resembled by SQL-Injections. Arguably, this would make the threat of SQL like Injections even worse, as users with no technical background are now able to query for confidential data as well. To mitigate that issue, we enforce a privacy-preserving AI, with the goal to never output SQL-queries that could disclose confidential data.

### 1.4 Gathering Data

TODO