# FullBrain: a Social E-learning Platform

Mirko Biasini[1], Vittorio Carmignani[1], Nicola Ferro[2], Panagiotis Filianos[1],
Maria Maistro[3], and Giorgio Maria di Nunzio[2]

[1] FullBrain, https://fullbrain.org/
{mirko, vitto, panos}@fullbrain.org
[2] University of Padua, Padova, Italy
{ferro,dinunzio}@dei.unipd.it
[3] University of Copenhagen, Copenhagen, Denmark
mm@di.ku.dk

**Abstract.** We present FullBrain, a social e-learning platform where students share and track their knowledge. FullBrain users can post notes, ask questions and share learning resources in dedicated course and concept spaces. We detail two components of FullBrain: a Social Information Retrieval (SIR) system equipped with query autocomplete and query autosuggestion, and a Leaderboard module to improve user experience. We analyzed the day-to-day users' usage of the SIR system, measuring a time-to-complete a request below 0.11s, matching or exceeding our UX targets. Moreover, we performed stress tests which lead the way for more detailed analysis. Through a preliminary user study and log data analysis, we observe that 97% of the users' activity is directed to the top 4 positions in the leaderboard.

**Keywords:** e-learning · social networks · social information retrieval · gamification · leaderboards

## 1 Introduction

E-learning platforms have been improving the quality of education in the digital sphere, with an increasing popularity over the past two decades. Nevertheless, the main approach has been to port the offline experience to its virtual analogue, as in the case of Massive Open Online Course (MOOC) platforms, which offer video lectures, note taking, quizzes, forums, etc. Well known examples of MOOC platforms are edX [2], Udemy [3] and Coursera [1]. However, MOOCs struggle to keep users engaged: approximately 88% of learners *do not return* after a first year of usage and on average only 8% of students per cohort are retained [67]. Moreover, we believe MOOC platforms to be structurally lacking. By mimicking the physical education experience, they miss the possibility of providing the novel experience that the digital medium can offer. A multitude of digital learning features has been proposed, such as personalization, gamification and social,

connectivist learning, however the attempts to align these elements have been sparse. In this context, we identify two main challenges.

First, the social component is a fundamental part of the learning process [7, 25, 48, 69], and when this is not provided by e-learning platforms, it is usually supplemented, in official university courses, by different services, as Slack [71], Teams [53], Piazza [63], Moodle [55], etc. However, many students consider these solutions inadequate and, when possible, they prefer to directly contact a peer they personally know and trust. Between each other, they use makeshift solutions like Facebook Groups or Messenger [20] and WhatsApp [21] group chats. Students there share the same context and can immediately assist on queries with understanding of the course requirements. The above, however, force users to combine a multitude of services to achieve all their goals. Second, a single centralized and reliable source of learning material is not available for online learners. Students often search and access to different sources through Web search engines, however they struggle in determining the quality of each resource, especially when they do not have enough knowledge to critically review them. The results are often ordered following an opaque process. The above require extended time to review learning resources before usage.

The necessity of social e-learning platforms able to address the above challenges has became evident during the COVID-19 pandemic, when there has been a large scale transfer of learning activities to online spaces. According to Unesco [77] over 90% of the world students are confined at home. Providing alternative learning solutions has become the top priority.

We present *FullBrain* a social e-learning platform which addresses these challenges. FullBrain is framed in the context of connectivism theory [26, 70], which represents learning as a network phenomenon influenced by technology and socialization. FullBrain offers open, dedicated spaces corresponding to university courses and concepts. It allows students to interact with these spaces, to post thoughts, ask questions and share, rate and comment online learning resources, tagging them with instructional metadata. These learning resources are organized in a crowdsourced ontological structure forming a reviewed *digital library*. In this paper, we first present FullBrain's features and then focus on two fundamental components of its architecture: Social Information Retrieval (SIR) System, that powers our search algorithms, and a user leaderboard, that encourages social connectivity in concept and course spaces. This paper is organised as follows: Section 2 details the related work; Section 3 introduces FullBrain's design, features and system's architecture; Section 4 and Section 5 present the SIR and the leaderboard components respectively; Section 6 showcases the prototyped solutions, which are evaluated in Section 7. Finally, Section 8 includes conclusions and future work.

## 2 Related Work

FullBrain belongs to the category of social e-learning platforms. Such efforts rarely have been part of literature and FullBrain stemmed from a few research

papers: Gil and Martin-Bautista [25] propose a connectivist, ontological knowledge support system with a focus on collectivist knowledge building; Liu et al. [48, 49] propose systems for collaborative relevance assessment with tasks as context; and Chen and Su [13] propose SAP, an ontology-based social learning platform with learning paths.

However none of these approaches have been adopted in a larger scale. edX [2], Coursera [1] and Udemy [3] are MOOC platforms that provide content, but no immediate social networking tools. In contrast, learning platforms that offer social learning, as Golden [19] and Expii [18], have a wiki approach to knowledge. Differently, FullBrain fully integrates the learning and social components in a single platform.

In addition to the social networking features, FullBrain offers a SIR System [41], which ranks results using both topical and user similarity. The user similarity is calculated based on the distance in the social graph between the searcher and the results [40, 42, 78]. We propose an architecture of an IRS specifically designed in the context of Online Social Networks [6] able to handle different entities, rather than only users (k-$n$ partite relationships [5]). To the best of our knowledge such architecture is novel.

Finally, FullBrain exploits gamification approaches through a leaderborad. Gamification has proven to be beneficial to increase user engagement in numerous contexts [16, 43, 44, 62]. Leaderboards are a workhorse of gamification [45] and are considered one of the key elements in game design [34]. According to Zichermann [23], leaderboards can follow two approaches: absolute and relative. Upon these, leaderboards emphasize continuous performance, status reporting, comparison, socialization, and competition [8, 14].

## 3   FullBrain System

In this section we present the elements that comprise FullBrain's user interface and system. We start with the description of the main features in Section 3.1, the underlying social graph in Section 3.2 and the overall architecture in Section 3.3.

### 3.1   Features

Six entities represent the core of FullBrain's offering: Posts, Streams, Sources, Courses, Concepts and User Profiles. We detail them next.

As in any regular Social Network, the users are able to create **posts** to share ideas. Also, users can post anonymously and specify if it is a question or a note. In the foreseeable future, they will be able to mark a question as 'resolved' if the issue was addressed. Posts are feature-heavy: users can like, comment and share any post they have access to.

**Streams** refer to collections of consecutive posts in a specific space: in the context of a course, concept or user profile page. The user home page is an example of stream, where users receive posts aggregated from streams of the entities they have *included* (followed).

An operational definition of **source** is: *a resource to learn one or multiple subjects which can be freely accessed online*. Simply, sources are URL links pointing to a destination website which provide learning content. Sources are shared by members of the FullBrain community. These resources combined with the user who shared them have a set of metadata that form the source card:

- *Name*, *affiliation* and *level* of the user who shared the resource. The level is an indicator of the user's reputation in the platform. It depends on, for example, user's interactions or obtained reviews on shared resources. This will be further addressed in Section 5;
- *Learning styles*, which can be: watching, reading, practise/exercise, listening, group activity or tutorial;
- *Title* which highlights the content of the source;
- *Instructions* on how to traverse the source. For instance, if the source is a video, the instructions may be: "watch from minute 2:03 to 5:40 to learn concept X";
- *Prerequisites*: concepts that users have to know in advance to understand the learning source. Clicking into a prerequisite results in opening the related concept page;
- Social stats: *star reviews* and *comments* allow to express an opinion on the source relevance for future viewers.

We refer to **course** as an academic course associated with its respective Institution. Examples can be 'Machine Learning' and 'Database Systems' at the University of Padua (Unipd) or 'Business Management' at the Copenhagen Business School (CBS). Each course has a dedicated space with group features: users can join and post thoughts and questions. Moreover, students can share learning resources helpful to pass the course/exam. The course's sources slightly differ from the ones in concepts: they do not have prerequisites, but tags, which allow to label the material based on the purpose: #lecture1, #exams or #exercises.

An operational definition of **concepts** can be as follows: *the smallest unit of knowledge, idea or process, which can not be divided further without becoming an other concept or losing utility for the task at hand*.

It is important to note that this is a soft definition, a guideline which is open for interpretation by FullBrain's members. Every FullBrain member can create a concept space, which allows post and resource sharing. In that respect interactions are very similar to Course spaces. Concepts are linked through learning resource prerequisites, forming crowdsourced ontologies. Moderating the validity of these ontologies is expected to be a social endeavor as well. A user experience workflow is not proposed currently and is matter of future research.

Each user has a **user profile** page, which includes the user name, institutional affiliation, description and picture. In addition it shows the list of includers (followers) and included users (followed users), the lists of the courses and concepts the user has included and a stream of their posts.
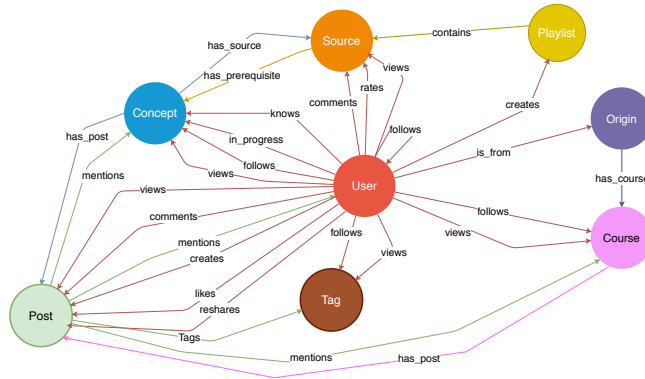
**Fig. 1.** FullBrain Social Graph with entities as nodes and related relationships as edges.

### 3.2 Social Graph

A social network can be seen as a graph [17, 56, 57]. Indeed, Elveny et al. [17] present the users as social actors which interact with each other creating relationships. In FullBrain, the relationships between users are: asymmetric, explicit, non-confirmed, regular and unsigned [29, 30].

The FullBrain social graph, shown in fig. 1, can be seen as an extended social graph with the $k$-partite definition [5], where nodes can be of different types. We recognize the main *entities* in Section 3.1 as nodes. In addition, we find *Origins* (institutions), *Tags* (labels for learning material and posts) and *Playlists* (public or private collections of favorite sources created by users). As edges, we can see all the possible relationships between the various entities (nodes).

### 3.3 Architecture

Figure 2 displays FullBrain's architecture. All virtual machines and databases are hosted in Azure [52]. Cloud services ensure scalability and continuity. We split FullBrain's architecture into three main parts: *front-end*, *back-end* and *databases*.

The front-end is the layer in direct contact with the user. When a user navigates through the FullBrain platform, it creates a series of requests. Those are forwarded from the front-end browser client to the back-end server. The back-end is in charge of handling those requests, and return back the expected results.

Multiple microservices compose the back-end [51]. To each Microservice (MS) we assign specific functionalities. MSs can communicate using gRPC [27] (a language independent protocol) using Protobuf serialization. Such architecture gives the possibility of developing each of the MSs in a different programming language to maximize the performance based on peculiarities and available libraries.

Finally, as we can see in fig. 2, each MS is connected to different databases, depending on their operations/calls. *PostgreSQL* [64] maintains most of Full-Brain's entities' fields data, and it ensures integrity between them. *Neo4j* [59],
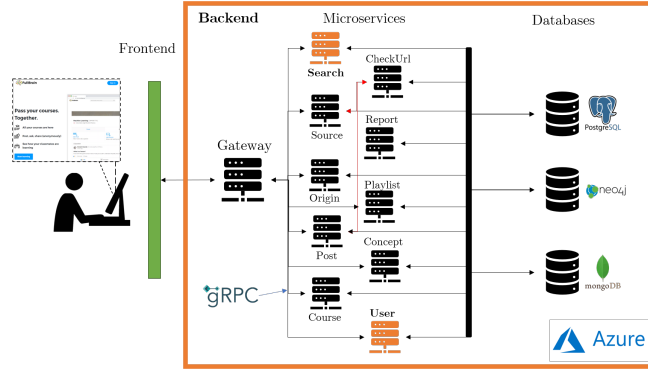
**Fig. 2.** FullBrain Architecture. In orange the Microservices (MS)s that contain the functions we describe in Section 4 and Section 5.

a graph database, manages the different entity relationships seen in Figure 1. *MongoDB* [54] is mostly utilised for logging of user's activities and requests.

## 4 Social Information Retrieval (SIR) System

Carmignani [11] describes FullBrain's SIR system, which resides in the Search MS (fig. 2). It offers three functionalities: *Search, Query Autocomplete (QAC)* and *Query Suggestion (QS)*. Search and QAC estimate relevance by computing the similarity between the query, the entities and the user, and then order the results accordingly to such value [46].

The overall similarity $S$ [40] for a query $q$ with respect to an entity $e$ and a user $u$ (searcher) is calculated as follow:

$$S(u, q, e) = \frac{\alpha S_T(q, e) + \beta S_U(u, e)}{\alpha + \beta} \in [0, 1] \quad \alpha, \beta \in \mathbb{R}^+ \tag{1}$$

where $S_T$ and $S_U$ are, respectively, *topical* and *user* similarity. We choose $\alpha = \beta = 1$ to give the same importance to the topical and user components.

The user similarity $S_U(u, e)$ in Equation (1) is computed as follows:

$$S_U(u, e) = 1 - \frac{d(u, e)}{\max d} \in [0, 1] \tag{2}$$

where $d$ is the length of the shortest path between $u$ and $e$ in the social graph (Section 3.2).

For efficiency reasons, we calculate an approximation of $d$ using a modified version of the landmark embedding method [6,35,50,65,73,75,78]. According to Lampe et al. [42] "people are more likely to search for other users who are part of their real life friends-network, rather than other members". Thus, we modify the standard algorithm by not saving distances greater than 3 (assuming them as infinite) as suggested by Vieira et. al. [78], since entities at such distance would

not be relevant for the user. This assumption has saved, in our case, more than 80% of landmarks-nodes stored distances.

The computation of the topical similarity $S_T(q, e)$ in Equation (1) differs for search and QAC. We denote it by $S_{Search}$ and $S_{QAC}$ respectively. The search function results include: concepts, users, courses, sources and posts. $S_{Search}$, is computed as a combination of partial [12, 37] and exact match as follows:

$$S_{Search}(q, e) = \frac{S_{Partial}(q, e) + S_{Exact}(q, e)}{2} \in [0, 1] \tag{3}$$

For partial matching $S_{Partial}(q, e)$, we employ q-grams [28,32,39,47,58,76]. Given the query $q$ and an entity name $n_e$ (e.g. user or concept name), we apply the same text operations on $q$ and $n_e$, retrieving their respective set of q-grams, $G_q$ and $G_{n_e}$:

$$S_{Partial}(q, e) = \frac{|G_q \cap G_{n_e}|}{|G_q \cup G_{n_e}|} \in [0, 1] \tag{4}$$

For exact matching $S_{Exact}(q, e)$, we use the Vector Space Model [46] and calculate $S_{Exact}(q, e)$ as the cosine similarity of the vector representations $\boldsymbol{q}$ and $\boldsymbol{e}$. Specifically for the latter, we use the union of $e$'s fields, like name and description (giving weights according to the fields' importance):

$$S_{Exact}(q, e) = cos(\boldsymbol{q}, \boldsymbol{e}_{union}) \in [0, 1] \tag{5}$$

FullBrain's QAC makes possible to retrieve a Qlist [72] of entities (users, concepts and courses) that are relevant for the user while typing the query itself [72]. As demonstrated in a variety of studies [9, 74, 80, 81], such features improve user satisfaction. The elements that populate the Qlist are ranked with the overall similarity in Equation (1), where the user similarity is computed as in Equation (2) and the topical similarity is $S_{QAC}(q, e) = S_{Partial}(q, e)$. We use only partial matching to not compromise the speed of retrieval required by QAC.

Finally, we describe the Query Suggestion (QS) component. We refer to QSs as the elements which populate the QList before typing any character into the query input form. QSs are either past queries (input for search function) or actual entities (direct link to the related web page). Such elements are retrieved based on the user's search history (more recent would result in a higher rank) without any similarity computation. We also suggest trending queries and entities. Those are retrieved using a modified version of the Most Popular Completion (MPC) model [10, 36, 68]; where a time filter is applied following the finding of [74, 79], which have shown that queries' popularity is heavily affected by the time.

## 5    Leaderboard

As previously mentioned, gamification can positively impact users' interactions and contributions. Thus, we designed and developed a leaderboard aiming to increase learners' engagement in FullBrain. Our platform offers a multitude of

| Action | Add | Share | Rate | Comment | Up Vote Comment | Add | Share | Rate | Comment | Up Vote Comment | Include |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Entity | Source | | | | | Post | | | | | User Cou. Con . |
| **Avg Effort** | 4.4 | 3.3 | 1.6 | 2.5 | 1.1 | 2.5 | 1.4 | 1.3 | 2.1 | 1.7 | 1.6 |
| **Avg Value** | 4.7 | 4.5 | 4.7 | 3.9 | 4 | 4.5 | 4.4 | 4.4 | 4.1 | 4 | 3.1 |
| **Sum** | 9.1 | 7.8 | 6.3 | 6.4 | 5.1 | 7.0 | 5.8 | 5.7 | 6.2 | 5.7 | 4.7 |

**Table 1.** Leaderboard points system actions' scores

actions, which can be performed in different contexts, such as courses and concepts. Combining different sets of actions and goals, a variety of leaderboards can be generated to address different motivational areas. Upon these, leaderboards provide learners with elements for reflection, comparison, motivation, socialization, and competition [8, 14].

*Points system and mechanics* To be created, leaderboards require a point system. The point system is a mechanism defining a set of criteria, which are used to assign points to the users. Through these points, a score is computed. Finally, users are ranked based on their final score.

As suggested by Jia et. al [38], leaderboards need to provide a sense of fairness to the users. To generate a points system as fair as possible, we categorized all available actions based on two parameters: effort and value. The former assesses the effort required to perform the action, inside and outside the platform. The latter is an evaluation of the value the performed action provides to the platform.

Among all the possible actions available in FullBrain, 11 were selected to build the points system. Precisely, *add*, *share*, *rate*, *comment*, *upvote comment* a source, *add*, *share*, *like*, *comment*, *upvote comment* a post, and *include user/ course/concept*. Several users were interviewed to assign an effort and a value score to each action. Scores were given between 1 and 5. The higher the score, the higher the effort/value. Next, these were averaged out to generate the final points. As a result, the final points system was built as shown in table 1. Every time a user performs one of these actions, the corresponding points are assigned to the user.

As described above, different types of leaderboards can be generated based on actions and contexts. To do so, several activity metadata are stored every time a user performs an action: the user performing the action, the performed action, the location where the action is performed (e.g. concept, course, user), the object undergoing the action (e.g. source, post, concept, etc), the date and time the action is performed, and the corresponding points. Finally, these activities are filtered based on one or more parameters, to generate the desired leaderboard.

To handle delete actions, such as deleting a post, analogous data are stored but with a negative score instead. Deleting actions affect only the user who performs the deleting action. Users who earned points by interacting with the deleted element remained unaltered. This is done to ensure fairness.

*Leaderboard design* To be effective, leaderboards need to be designed based on the audience and the context [15, 33, 38, 45]. In FullBrain, concepts have a larger and diverse audience compared to courses. Indeed, as courses refer to

a specific university, they are generally restricted to those students who are currently taking that course. Conversely, concepts are followed by learners all around the world. To determine the most appropriate design for each scenario, interviews, A/B tests [61] and thinking-aloud usability evaluations [31] were conducted aiming to determine four main points.

– **Leaderboard design:** four different designs were tested: an absolute, a relative, and two hybrids. The absolute solution focuses on the top players, whereas the relative is centered on the active user [23,38]. On top of these, we proposed two hybrid designs, which combine both approaches to exploit their advantages. Each design affects different motivational areas like reflection, competition, or socialization.
– **Leaderboard items:** which and how information should be displayed in the leaderboards. Here, focus was directed towards users' scores, to determine whether showing points is more effective or not.
– **Time:** Leaderboard points can be calculated based on user activities included in a time window of the current week, month, semester etc. We should determine which time windows provide valuable information to users.
– **Leaderboard type:** identify which type of leaderboards are more interesting for users. For example, we can have *Top Contributor*, *Top Reviewer* or *Top Influencer* leaderboards. Each ranks users based on points collected from a different set of activities.

The results of these experiments led to the final designs described in the following section. Finally, a specific leaderboard is generated for each concept and course. Note, the way we store data allows creating general leaderboards, comprising groups of concepts and/or courses. For example, the top contributors of a specific university or of a big subject like Machine Learning. More research has been conducted on that subject by Biasini [4].

## 6 Prototype

**Social Information Retrieval System -** Figure 3 displays the results of using FullBrain's SIR in the FullBrain main search bar. Figure 3A shows the QS, where five past queries and entities are suggested together with as many popular ones. Figure 3B instead, shows a QAC output and how the use of trigrams permits to retrieve "Vittorio Carmignani" despite the input typos. Lastly, Figure 3C,D,E show three different example of the search function. In *C*, $q=$"pca", we can recognize the concept, together with posts and sources. In *D*, $q=$"dtu" (Technical University of Denmark), the results are retrieved even if not present in the entities'names. This is because we are using additional fields in the search process. Ultimately, in *E*, $q=$"Vittorio Karmignani", despite the typo we can still retrieve the results thanks to the use of partial matching in the search function. A working example of the SIR system is available online[4].
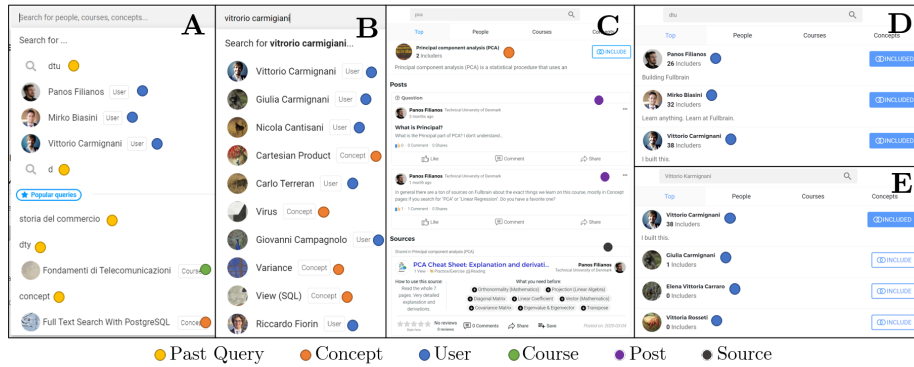
---

[4] `https://youtu.be/6YOm-xobCzA`

**Fig. 3.** Example of QS (A), QAC (B) and Search Results (C,D,E) for the user "Vittorio Carmignani". Moreover, the legend underneath highlights the different entities.

**Leaderboard -** We propose two different hybrid solutions as a result of user interviews, A/B tests and usability evaluations: *Hybrid-Absolute Design* and *Hybrid 50-50 Design* (fig. 4). We use the hybrid-Absolute design for concepts: an absolute leaderboard shows the top 10 users and the active user is always shown at the bottom of the leaderboard, regardless of his/her ranking. For courses we use the hybrid 50-50 design, a novel approach combining an absolute leaderboard, on the top, with a relative leaderboard centered on the active user, on the bottom.

In both solutions, users can access top users' profiles and universities, as well as include them in their home stream. A time filter allows users to change the time window used to build the leaderboard. In addition, users can change the type of the leaderboard choosing between: Top Contributor, which ranks users accounting for all types of actions, and Top Responder, which ranks users accounting just for the frequency of their answers and the quality of their comments. Both solutions are partially implemented in the platform and we are currently working to complete them.

## 7   Evaluation and Results

**Social Information Retrieval System -** Carmignani [11] focused on the efficiency of FullBrain's IRS. Based on J. Nielsen's studies [60], he set that the time to complete a **QS**, **QAC** and **Search** request should be, respectively, **0.1s** (instantly), in **[0.1-1]s** (small delay without interrupting user's flow of thought) and **[1-1.5]s** (it is not an interactive function). Hence, we have logged the time to complete a QS, QAC and Search requests on FullBrain from all user clients using the platform for 3 weeks. The results are shown in the "Day-to-day" usage columns in table 2. We can see that **all three functions were within the bounds set**. Therefore, we performed stress tests to test the limits of the architecture. Particularly we made $1000/n$ requests from $n$ different users in
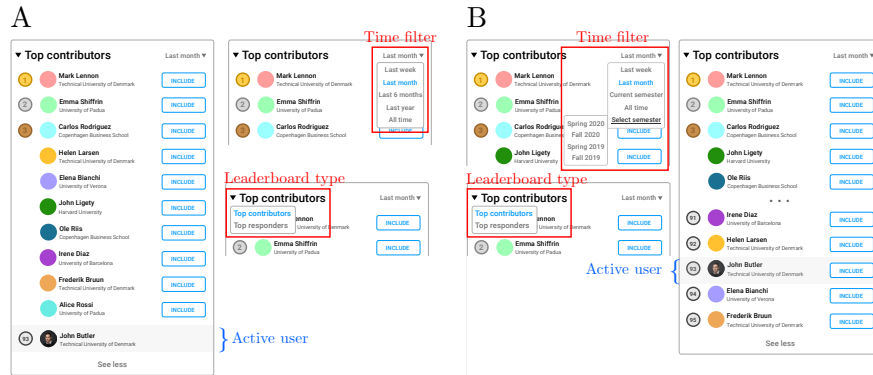
**Fig. 4.** Leaderboard designs for Concept and Course spaces. (A) Hybrid-Absolute design for concepts. (B) Hybrid 50-50 design for courses.
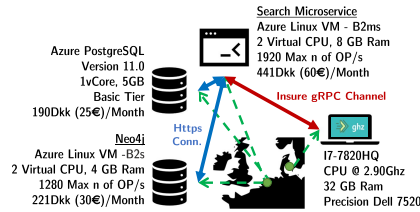


**Fig. 5.** Setup for Stress Tests and components specs.

parallel (with $n \in \{1, 2, 4, 8, 16, 32, 64\}$) using *ghz* [24]. The test setup is shown in fig. 5. The functions were applied on a dataset having a total of 5724 entities and 21512 relationships. The results are shown in the stress test columns in table 2. All functions show a similar performance patterns: they all break the set bounds after $n = 4$. In contrast, it is important to remark here that, by using the website, it is not possible to reproduce this kind of loaded environment (with only these number of users): there would be a delay between two users' requests due to the UI and the physical action of submitting them. Although this delay can be minimal, it can be still valuable since in normal conditions a request is answered in 0.1s. In contrast, we know that this delay is zero in case of stress test (since the requests are programmatically made). Thus, we believe that the system can support far more than 64 parallel users using the platform (admitting that the tests machine utilized are not suitable for a production environment). Indeed, between user request submission and its answer computation, there is a whole structure of microservices and communications that takes place. If the microservices are the problem, then we could create clones and balance the load of requests. If the problem is the computation, we need to know if it is due to physical limits (machine not performant enough) or the actual implementation. Thus, further analysis would be required to understand exactly in which step the bottleneck is of the overall process.

| | Day-to-Day | | Stress Tests | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Count | Avg Time | Count | Avg Time (s) for a Request with $n$ concurrent users | | | | | | |
| | | | | 1 | 2 | 4 | 8 | 16 | 32 | 64 |
| **QS** | 8743 | 0.06s | 1000 | 0.039 | 0.078 | 0.159 | 0.3444 | 0.614 | 1.191 | 2.390 |
| **QAC** | 7599 | 0.11s | 1000 | 0.506 | 1.323 | 0.920 | 1.912 | 4.769 | 7.611 | 15.029 |
| **Search** | 512 | 0.10s | 1000 | 0.234 | 0.555 | 0.594 | 1.751 | 2.135 | 1.052 | 7.754 |

**Table 2.** Efficiency Study on FullBrain's SIR based on day-to-day usage on the platform and conducted stress tests.

**Leaderboard -** During the experiments conducted to determine the final leaderboard designs, 73% of 15 testers preferred the *hybrid-absolute* design for concepts. When asked to justify such a choice, participants claimed that in broader contexts like concepts, they primarily care about top users. They only want to see how top users learn and get inspired from them. Conversely, opposite results were found for courses. Indeed, 75% chose the *hybrid 50-50* solution. Here, users explained that since they have real relationships with other students, they also want to see similar classmates. This was justified for two reasons. Firstly, users want to assess themselves compared to the rest of the class. Secondly, users feel more comfortable to approach similar students rather than top ones [4]. To validate leaderboards' effect on users' engagement, we performed a longitudinal validation. We collected users activities made through the leaderboards, for a total of 30 days. Even though the project constraints did not allow us to produce evidence with statistical significance, preliminary results suggest that users' activities are mainly focused on top positions. In fact, 55% of the activities observed in the leaderboards occurred in the best users (1st position in the table). Moreover, more than 97% happened within the first 4 positions.

# 8 Conclusions and Future work

FullBrain has shown a lot of promise to provide a digital home for learners. From the release of the social features on FullBrain (October 2020) until the current publication we have experienced an increase of 146% in new registrations, totalling at approximately 500 users. In addition, we have described FullBrain's Social Information Retrieval System (SIR) and leaderboards. The SIR provides the following functions: Search, Query Autocomplete and Query Suggestions (no query input). These permit to explore platform's content, search through multiple entities while using the social graph to rank the results themselves. Moreover, the architecture is able to build the related user's responses in less than 0.11s. For the Leaderboards, we propose a hybrid-absolute design in concepts and a hybrid 50-50 design for courses, based on user interviews. Through preliminary analysis we recognise that Leaderboards produce user activity focused on the top ranking members with 97% of activity directed to the top 4 positions.

Larger user numbers allow us to experiment with Machine Learning and Deep Learning systems. Previous work by Filianos [22] has attempted to use

Multigated Mixture of Expert (MMoE) [82] and Mixture of Sequential Experts (MoSE) [66] models to rank posts in our home page stream. As our user base grows, we consider FullBrain to be very fertile ground for learning analytics and user activity dataset research.

## References

1. Coursera: Build Skills with Online Courses from Top Institutions. `https://www.coursera.org/`, Accessed: 2020-12-12
2. edX: Free Online Courses by Harvard, MIT, & more. `https://www.edx.org/`, Accessed: 2020-12-12
3. Udemy: Online Courses - Learn Anything, On Your Schedule. `https://www.udemy.com/`, Accessed: 2020-12-12
4. Biasini, M.: Designing analytics tools for decision-making in a crowd-sourced educational platform for STEM students. Master's thesis, University of Padua (12 2020)
5. Bouadjenek, M.R., Hacid, H., Bouzeghoub, M.: Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms. Information Systems **56**, 1–18 (2016). https://doi.org/10.1016/j.is.2015.07.008, `http://dx.doi.org/10.1016/j.is.2015.07.008`
6. Boyd, D.M., Ellison, N.B.: Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication **13**(1), 210–230 (2007). https://doi.org/10.1111/j.1083-6101.2007.00393.x, `https://academic.oup.com/jcmc/article/13/1/210/4583062`
7. Burke, M., Marlow, C., Lento, T.: Feed me: Motivating newcomer contribution in social network sites. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 945–954. CHI '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1518701.1518847, `https://doi.org/10.1145/1518701.1518847`
8. Butler, C.: The effect of leaderboard ranking on players' perception of gaming fun. Online Communities and Social Computing. 5th International Conference, Ocsc 2013 Held As Part of Hci International 2013. Proceedings. Lncs 8029 pp. 129–36, 129–136 (2013). https://doi.org/10.1007/978-3-642-39371-6_15
9. Cai, F., De Rijke, M.: A survey of query auto completion in information retrieval. Foundations and Trends in Information Retrieval **10**(4), 273–363 (2016). https://doi.org/10.1561/1500000055
10. Cai, F., Liang, S., De Rijke, M.: Time-sensitive personalized query auto-completion. In: CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management. pp. 1599–1608 (2014). https://doi.org/10.1145/2661829.2661921, `http://dx.doi.org/10.1145/2661829.2661921.`
11. Carmignani, V.: Design and Development of an Information Retrieval System on an Educational Social Network for STEM students. Master's thesis, University of Padua (12 2020)
12. Chaudhuri, S., Kaushik, R.: Extending autocompletion to tolerate errors (2009). https://doi.org/10.1145/1559845.1559919, `http://www.amazon.com`
13. Chen, Y., Su, C.: Ontology and learning path driven social e-learning platform. In: 2016 International Symposium on Computer, Consumer and Control (IS3C). pp. 358–361 (2016). https://doi.org/10.1109/IS3C.2016.99

14. Christy, K.R., Fox, J.: Leaderboards in a virtual classroom: A test of stereotype threat and social comparison explanations for women's math performance. Computers and Education **78**, 66–77 (2014). https://doi.org/10.1016/j.compedu.2014.05.005

15. Codish, D., Ravid, G.: Personality based gamification: How different personalities percive gamification. Ecis 2014 Proceedings - 22nd European Conference on Information Systems (2014)

16. Costa, J.P., Wehbe, R.R., Robb, J., Nacke, L.E.: Time's up: Studying leaderboards for engaging punctual behaviour. Acm International Conference Proceeding Series pp. 26–33 (2013). https://doi.org/10.1145/2583008.2583012

17. Elveny, M., Syah, R., Elfida, M., Nasution, M.K.: Information Retrieval on social network: An Adaptive Proof. IOP Conference Series: Materials Science and Engineering **300**(1) (2018). https://doi.org/10.1088/1757-899X/300/1/012055

18. Expii: Expii — Learn Creatively (2020), `https://www.expii.com/`

19. Expii: Golden — The intelligent knowledge base (2020), `https://golden.com/`

20. Facebook: Messenger, `https://www.messenger.com/`

21. Facebook: WhatsApp, `https://www.whatsapp.com/?lang=it`

22. Filianos, P.: A novel Deep Learning ranking model to order posts on a social e-learning platform. Master's thesis, Technical University of Denmark (12 2020)

23. Gabe Zichermann, C.C.: Gamification by design: Implementing game mechanics in web and mobile apps. O'Reilly Media (2011)

24. Ghz: ghz · Simple gRPC benchmarking and load testing tool, `https://ghz.sh/`

25. Gil, R., Martin-Bautista, M.: A novel integrated knowledge support system based on ontology learning: Model specification and a case study. Knowledge-Based Systems **36**, 340 – 352 (2012). https://doi.org/https://doi.org/10.1016/j.knosys.2012.07.007, `http://www.sciencedirect.com/science/article/pii/S0950705112001955`

26. Goldie, J.: Connectivism: A knowledge learning theory for the digital age? Medical Teacher **38**, 1064 – 1069 (2016)

27. Google: gRPC – A high-performance, open source universal RPC framework, `https://grpc.io/`

28. Gravano, L., Ipeirotis, P.G., Jagadish, H.V., Koudas, N., Muthukrishnan, S., Pietarinen, L., Srivastava, D.: Using q-grams in a DBMS for Approximate String Processing. Tech. Rep. 4 (2001), `http://dblp.uni-trier.de/db/journals/debu/debu24.html\#GravanoIJKMPS01`

29. Guy, I.: People recommendation on social media. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10100 LNCS, pp. 570–623. Springer Verlag (2018). https://doi.org/10.1007/978-3-319-90092-6_15

30. Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E.: Social media recommendation based on people and tags (2010). https://doi.org/10.1145/1835449.1835484, `www.stumbleupon.com`

31. H. Birns, J., A. Joffre, K., F. Leclerc, J., Andrews Paulsen, C.: Getting the whole picture: Collecting usability data using two methods – concurrent think aloud and retrospective probing (2016). https://doi.org/10.1.1.111.5466

32. Hall, P.A., Dowling, G.R.: Approximate String Matching. ACM Computing Surveys (CSUR) **12**(4), 381–402 (12 1980). https://doi.org/10.1145/356827.356830, `https://dl.acm.org/doi/10.1145/356827.356830`

33. Hamari, J., Koivisto, J., Sarsa, H.: Does gamification work? - a literature review of empirical studies on gamification. Proceedings of the Annual Hawaii

International Conference on System Sciences pp. 6758978, 3025–3034 (2014). https://doi.org/10.1109/HICSS.2014.377

34. Hickman, T.: Book review : Reeves, b., & read, j.l. (2009). total engagement: Using games and virtual worlds to change the way people work and businesses. harvard business school press. Computers in Human Behavior **26**(6), 1276–1277 (2010). https://doi.org/10.1016/j.chb.2010.03.035

35. Hotz, M., Chondrogiannis, T., Wörteler, L., Grossniklaus, M.: Experiences with implementing landmark embedding in Neo4j. Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (2019). https://doi.org/10.1145/3327964.3328496, `https://doi.org/10.1145/3327964.3328496`

36. Hu, Y., Xiao, C., Ishikawa, Y.: Context-Sensitive Query Auto-Completion with Knowledge Base. Tech. rep. (2018)

37. Ji, S., Li, G., Li, C., Feng, J.: Efficient interactive fuzzy keyword search (2009). https://doi.org/10.1145/1526709.1526760

38. Jia, Y., Liu, Y., Yu, X., Voida, S.: Designing leaderboards for gamification: Perceived differences based on user ranking, application domain, and personality traits. Conference on Human Factors in Computing Systems - Proceedings **2017-**, 1949–1960 (2017). https://doi.org/10.1145/3025453.3025826

39. Kim, M.S., Whang, K.Y., Lee, J.G.: n-Gram/2L-approximation: A two-level n-gram inverted index structure for approximate string matching. Tech. Rep. 6 (2007)

40. Kirchhoff, L., Stanoevska-Slabeva, K., Nicolai, T., Fleck, M.: Using Social Network Analysis To Enhance Information Retrieval Systems. In: Proceedings of the 5th Conference on Applications of Social Network Analysis (sep 2008)

41. Kirsch, S.M., Gnasa, M., Cremers, A.B.: Beyond the Web: Retrieval in social information spaces. Tech. rep. (2006). https://doi.org/10.1007/11735106_9

42. Lampe, C., Ellison, N., Steinfield, C.: A face(book) in the crowd: Social Searching vs. social browsing. Tech. rep. (2006). https://doi.org/10.1145/1180875.1180901

43. Landers, R.N., Bauer, K.N., Callan, R.C.: Gamification of task performance with leaderboards: A goal setting experiment. Computers in Human Behavior **71**, 508–515 (2017). https://doi.org/10.1016/j.chb.2015.08.008

44. Landers, R.N., Landers, A.K.: An empirical test of the theory of gamified learning: The effect of leaderboards on time-on-task and academic performance. Simulation and Gaming **45**(6), 769–785 (2014). https://doi.org/10.1177/1046878114563662

45. Leung, W.: How do one's peers on a leaderboard affect oneself? Conference on Human Factors in Computing Systems - Proceedings pp. 1–11 (2019). https://doi.org/10.1145/3290605.3300397

46. Levene, M.: Search Engines: Information Retrieval in Practice. Tech. Rep. 5 (2011). https://doi.org/10.1093/comjnl/bxq039

47. Li, G., Feng, J., Li, C.: Supporting search-as-you-type using SQL in databases. IEEE Transactions on Knowledge and Data Engineering **25**(2), 461–475 (2013). https://doi.org/10.1109/TKDE.2011.148, `http://www.ijcttjournal.org`

48. Liu, D.R., Wu, I.C.: Collaborative relevance assessment for task-based knowledge support. Decision Support Systems **44**(2), 524 – 543 (2008). https://doi.org/https://doi.org/10.1016/j.dss.2007.06.015, `http://www.sciencedirect.com/science/article/pii/S0167923607000991`

49. Liu, D.R., Wu, I.C., Yang, K.S.: Task-based k-support system: disseminating and sharing task-relevant knowledge. Expert Systems with Applications **29**(2), 408 – 423 (2005). https://doi.org/https://doi.org/10.1016/j.eswa.2005.04.036, `http://www.sciencedirect.com/science/article/pii/S0957417405000898`

50. Madkour, A., Aref, W.G., Rehman, F.U., Rahman, M.A., Basalamah, S.: A Survey of Shortest-Path Algorithms. Tech. rep. (2017), `http://arxiv.org/abs/1705.02044`

51. Mazzara, M., Meyer, B.: Present and ulterior software engineering (2017). https://doi.org/10.1007/978-3-319-67425-4, `https://doi.org/10.1007/978-3-319-67425-4`

52. Microsoft: What is Azure – Microsoft Cloud Services — Microsoft Azure (2019), `https://azure.microsoft.com/en-us/overview/what-is-azure/`

53. Microsoft: Microsoft Teams — Group Chat, Team Chat & Collaboration (2020), `https://www.microsoft.com/en-gb/microsoft-365/microsoft-teams/group-chat-software`

54. MongoDB: The most popular database for modern apps — MongoDB (2020), `https://www.mongodb.com/`

55. Moodle: Moodle - Open-source learning platform — Moodle.org (2018), `https://moodle.org/`

56. Nasution, M.K., Hardi, M., Syah, R.: Mining of the social network extraction. Journal of Physics: Conference Series **801**(1) (2017). https://doi.org/10.1088/1742-6596/801/1/012020

57. Nasution, M.K.: Social network mining (SNM): A definition of relation between the resources and SNA. International Journal on Advanced Science, Engineering and Information Technology **6**(6), 975–981 (2016). https://doi.org/10.18517/ijaseit.6.6.1390

58. Navarro, G., Sutinen, E., Tarhio, J.: Indexing text with approximate q-grams. In: Journal of Discrete Algorithms. vol. 3, pp. 157–175. Elsevier (jun 2005). https://doi.org/10.1016/j.jda.2004.08.003

59. Neo4j: Neo4j Graph Platform – The Leader in Graph Databases (2020), `https://neo4j.com/`

60. Nielsen, J.: Response Time Limits: Article by Jakob Nielsen (1993), `https://www.nngroup.com/articles/response-times-3-important-limits/`

61. Nielsen, J.: Putting a/b testing in its place (2019), `https://www.nngroup.com/articles/putting-ab-testing-in-its-place/`

62. Ortiz-Rojas, M., Chiluiza, K., Valcke, M.: Gamification through leaderboards: An empirical study in engineering education. Computer Applications in Engineering Education **27**(4), 777–788 (2019). https://doi.org/10.1002/cae.12116

63. Piazza: Piazza . Ask . Answer . Explore . Whenever. (2017), `https://piazza.com/`

64. PostgreSQL: PostgreSQL: The world's most advanced open source database (2013), `https://www.postgresql.org/http://www.postgresql.org/`

65. Potamias, M., Bonchi, F., Castillo, C., Gionis, A.: Fast shortest path distance estimation in large networks (2009). https://doi.org/10.1145/1645953.1646063

66. Qin, Z., Cheng, Y., Zhao, Z., Chen, Z., Metzler, D., Qin, J.: Multitask Mixture of Sequential Experts for User Activity Streams, p. 3083–3091. Association for Computing Machinery, New York, NY, USA (2020), `https://doi.org/10.1145/3394486.3403359`

67. Reich, J., Ruipérez-Valiente, J.A.: The mooc pivot. Science **363**(6423), 130–131 (2019). https://doi.org/10.1126/science.aav7958, `https://science.sciencemag.org/content/363/6423/130`

68. Shokouhi, M.: Learning to personalize query auto-completion (2013). https://doi.org/10.1145/2484028.2484076

69. Shum, S.B., Ferguson, R.: Social learning analytics. Journal of Educational Technology & Society **15**(3), 3–26 (2012), `http://www.jstor.org/stable/jeductechsoci.15.3.3`

70. Siemens, G.: Connectivism: A learning theory for the digital age. International Journal of Instructional Technology and Distance Learning **2** (2004)
71. Slack: Where work happens — Slack (2019), `https://slack.com/`
72. Smith, C.L., Gwizdka, J., Feild, H.: The use of query auto-completion over the course of search sessions with multifaceted information needs. Information Processing and Management **53**(5), 1139–1155 (2017). https://doi.org/10.1016/j.ipm.2017.05.001, `http://dx.doi.org/10.1016/j.ipm.2017.05.001`
73. Sommer, C.: Shortest-path queries in static networks (2014). https://doi.org/10.1145/2530531, `http://dx.doi.org/10.1145/2530531`
74. Tahery, S., Farzi, S.: Customized query auto-completion and suggestion — A review (2020). https://doi.org/10.1016/j.is.2019.101415, `https://doi.org/10.1016/j.is.2019.101415`
75. Tretyakov, K., Armas-Cervantes, A., García-Bãuelos, L., Vilo, J., Dumas, M.: Fast fully dynamic landmark-based estimation of shortest path distances in very large graphs (2011). https://doi.org/10.1145/2063576.2063834
76. Ukkonen, E.: Approximate string-matching with q-grams and maximal matches. Theoretical Computer Science **92**(1), 191–211 (1 1992). https://doi.org/10.1016/0304-3975(92)90143-4
77. UNESCO: COVID-19: Learning Cities on the Front Line (2020), `https://en.unesco.org/news/covid-19-learning-cities-front-line`
78. Vieira, M.V., Golgher, P.B., Fonseca, B.M., De Reis, D.C., Damazio, R., Ribeiro-Neto, B.: Efficient search ranking in social networks (2007). https://doi.org/10.1145/1321440.1321520
79. Wang, Y., Ouyang, H., Deng, H., Chang, Y.: Learning Online Trends for Interactive Query Auto-Completion. IEEE Transactions on Knowledge and Data Engineering **29**(11), 2442–2454 (11 2017). https://doi.org/10.1109/TKDE.2017.2738639
80. White, R.W., Marchionini, G.: Examining the effectiveness of real-time query expansion. Information Processing and Management **43**(3), 685–704 (may 2007). https://doi.org/10.1016/j.ipm.2006.06.005
81. Zhang, A., Goyal, A., Kong, W., Deng, H., Dong, A., Chang, Y., Gunter, C.A., Han, J.: AdaQAC: Adaptive query auto-completion via implicit negative feedback. SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 143–152 (2015). https://doi.org/10.1145/2766462.2767697, `http://dx.doi.org/10.1145/2766462.2767697`
82. Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., Chi, E.: Recommending what video to watch next: A multitask ranking system. In: Proceedings of the 13th ACM Conference on Recommender Systems. p. 43–51. RecSys '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3298689.3346997, `https://doi.org/10.1145/3298689.3346997`