Master Team Project WiSe 2021

pick-a-tutor.com

GDSD Project Team A

Hochschule Fulda AI Department

18-Nov-2021

Team Members

Name	Email ID	Special role
Alexander Wiegel	alexander.wiegel@ai.hs-fulda.de	Team Lead
Rakesh Kumar	rakesh.kumar@ai.hs-fulda.de	Backend Lead
Abdullah Butt	abdullah.butt@ai.hs-fulda.de	
Chahat Soni	chahat.soni@ai.hs-fulda.de	Frontend Lead
Maksim Kanushin	maksim.kanushin@ai.hs-fulda.de	Github Master
Ziad Khaled	ziad.khaled-mohamed@ai.hs-fulda.de	

Revision History

Revision	Date Submitted	Date Revision	Changes

Table of contents

1. Executive Summary

Some students need more attention for their academic development than others. For that they seek help on the internet and in their hometowns, spending valuable time on searching for the right person to aid them, when instead they could get help from other students or real teachers that have a direct connection to the students' school or university. Pick-A-Tutor will do exactly this.

Pick-A-Tutor is a website where students from Fulda University of Applied Sciences can find tutors that are registered there. Students can access it from any device. Two types of users can register on the website namely tutors and students. Only users with the email suffix *hs-fulda.de* will be allowed to register on this website to ensure that only valid users of Fulda can register here. Website Administrator users will be predefined and do not have to register on the website. The language of the website will be English.

The tutors will have to upload their CV, picture, enter pricing and mention the subjects they can teach. They can also upload images, videos, audio, and post comments. After a tutor has signed up, the Administrator user will have a look at the tutor's profile and all uploaded content. If an Administrator deems the uploaded content as inappropriate, he/she can delete the inappropriate content and post a comment against the tutor profile which is only visible to the tutor. There will be four flags against the tutor's profile namely pending, approved, rejected, and blocked which the Administrator user can apply at any time. If the flag is rejected, the tutor can read the comments added by Administrator and then apply the changes or new uploads mentioned in comments by Administrator and then re-submit his profile. The status of the tutor will become pending again and can be approved or rejected again by the Administrator who can see the earlier comments posted by him/her earlier as well against the profile. If the Administrator blocks the tutor, the tutor can't update his profile and will never be listed on the website.

If the Administrator approves the tutor profile, only then the respective tutor will be listed on the website and can be searched and contacted.

A student can contact a tutor only by messaging service built in the website. The email addresses of students and tutors will not be visible on the website. A student can see the list of all registered approved tutors. A student can search a tutor by text search like name and can also apply filters of subject to learn, pricing and number of years of experience of tutor. The search result will list tutors. A student can click on any tutors in the list to be able to see the tutor's profile and pricing.

There will be no payment service included in the website. After a student contacts a tutor using the website's messaging service, if both parties agree to tutoring, the student can send a request from the website to enroll against the tutor. The tutor can accept or reject the student request to enroll under his tutorship. If the teacher approves, the student will be linked to the teacher as enrolled to him/her. It is understood that they have mutually agreed on the tuition fee and they'll manage the payments outside the website.

Privacy of all users of the website will be protected and privacy policy will be provided in the privacy policy link to all users. At time of sign up, the users will check the

checkbox next to the privacy policy link button that they have read and understand the privacy policies.

2. Personae and main Use Cases

A. Searching, Reviewing and Contacting the available tutors;

B. Uploading tutor information (CV, comments, images, video, audio, pricing etc.);

C. Functions for site Admin who will be required to approve each uploaded tutor info before it can go live, as well as to delete inappropriate items or users.

D. Compare the prices of selected tutors

3. List of main data items and entities

1. Students: A group of users who can view available courses and tutors, start conversations with tutors, register to courses, and review courses they have taken.

2. Tutors: A group of users who can teach courses, upload files corresponding to their taught courses, and reply to student messages.

3. Administrator Users: A group of users who moderate the website; they can start conversations with any other users, accept or reject applicant tutors, accept or reject tutors courses, view and act on reported reviews, and lock or completely ban rules violators users.

4. Course (taught by tutors): The set of material explained by tutors to students.

5. Pricing: The course prices set by each tutor for his taught courses.

6. Review: The rating and comment given by the student for the tutor after taking a course.

7. Approval:

8. Messages: The communication method by different users supported by the website

9. Profile photo: An image file that shows a user face.

10. Files: Uploaded files by tutors for their taught courses

4. Initial list of functional requirements

General:

01. Anonymous user or Normal Fulda University Student or Administrator should be able to browse through the available tutors and corresponding courses offered on the website.

02. Anonymous user or Normal Fulda University Student or Administrator should be able to browse through the available courses and corresponding tutors who are offering those courses.

03. Any Fulda university Student (whoes email suffix string has "hs-fulda.de") should be allowed to register with details such as First Name, Last Name, Email Id and Password.

04. Any Fulda University Student or Employees (whoes email suffix string has "hs-fulda.de") should be allowed to register as Tutor with details such as First Name, Last Name, Email Id and Password.

05. Registered users should be able to login on this website based on their assigned role (Student or Tutor or Administrator).

Student:

06. A registered Fulda University User should be able to view the reviews about a tutor on the website.

07. Any registered Fulda University User or Tutor should be able to reset their password by clicking on the forgot password option on the sign in screen.

08. Any registered Fulda University User or Tutor who would like to delete their account from this web site should be allowed to delete if logged in.

09. A registered User or Student should be able to contact the Tutor or Administrator via message on the website, these messages will stay within this website for Tutor or Administrator to view or reply.

10. Registered Users or Students should be able to write reviews about the tutor. These reviews will be shown on the tutor's page.

11. Registered Users or Students should be able to delete their reviews.

Tutor:

12. A registered Tutor should be allowed to add, update and delete tutor name, description, Additional Information and Pricing if logged in.

13. A registered tutor should be able to upload or delete the CV in his profile.

14. A registered tutor should be able to upload or delete the profile image.

15. A registered tutor should be able to upload or delete the files (these files can be Audio, Video, image files or documents). These files should be available for download to the students.

16. Registered tutor should be able to reply to the messages received on the website.

17. Registered tutor should be able to write new messages or reply to the messages received from the Administrator.

18. Registered tutor should be able to report a review to the Administrator written by the Fulda University Student in case it is inappropriate.

Administrator:

19. Administrator should be able to write new messages or reply messages of any Fulda University Student or Tutor.

20. Administrator should be able to approve a Fulda University Student or Tutor after verifying their details. In case of a tutor, the Administrator should be able to verify profile information including Images, Videos and CV.

21. Administrator should be able to block or unblock a Fulda University Student or Tutor on need basis.

22. Administrator should be allowed to delete a registered Tutor or Fulda University Student account.

23. Administrator should be able to view or delete the reported reviews.

5. List of non-functional requirements

01. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in Milestone 0. Application delivery shall be from the chosen cloud server.

02. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.

03. All or selected application functions must render well on mobile devices

04. Data shall be stored in the database on the team's deployment cloud server.

05. No more than 50 concurrent Students shall be accessing the application at any time.

06. Privacy of Students shall be protected and all privacy policies will be appropriately communicated to the Students.

07. The language used shall be English (no localization needed).

08. Application shall be very easy to use and intuitive.

09. Application should follow established architecture patterns.

10. Application code and its repository shall be easy to inspect and maintain.

11. Google analytics shall be used (optional for Fulda teams).

12. No email clients shall be allowed.

13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

14. Site security: basic best practices shall be applied (as covered in the class) for main data items.

15. Application shall be media rich (images, video etc.). Media formats shall be standard as used in the market today.

16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.

17. For code development and management, as well as documentation like formal milestones required in the class, each team shall use their own GitHub to be set-up by class instructors and started by each team during Milestone 0.

18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "Fulda University of Applied Sciences Software Engineering Project, Fall 2021 For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive analysis

Feature	Us	Chegg study	italki	Fiverr	erstenachhilfe. de	nachhilfe-team. net/	findmytutor.com
Only Fulda users	✔	✘	✘	✘	✘	✘	✘
In app messaging	✔	✘	✔	✔	✔	✘	✔
Tutor reviews	✔	✘	✘	✔	✘	✔	✘
Search for courses	✔	✔	✔	✔	✔	✔	✘
Search for tutors	✔	✘	✔	✘	✘	✘	✘
Extended filters	✘	✔	✔	✘	✔	✘	✔
Search by PLZ	?	✘	✔	✘	✔	✔	✘
Register as a tutor	✘	✔	✔	✔	✘	✘	✘
Report abuse	✔	✔	✔	✔	✘	✘	✔
Diverse courses	✔	✔	✘	✔	✔	✔	✔
Pay as you go	✔	✘	✔	✔	✔	✔	✔

The most distinctive feature of Pick-a-Tutor is that it is limited to Hochschule Fulda University Students only, whilst the others are publicly available. Most of the services lack the functions provided by Pick-A-Tutor: some of them Pick-a-tutor allow tutor reviews, but most of them do not. Most of the services also do not have an option to search by tutors if you know their name, but Pick-a-Tutor provides this feature since there is quite a small

community and Hochschule Fulda University Students might recommend tutors they communicate with to the other students.

7. High-level system architecture and technologies used

System architecture

This web application uses three tier client server architecture. The client's (laptop, computer, mobile, tablet) internet browser is a presentation layer, whereas application and database will be hosted on the server.

Technologies used

Supported browsers: Chrome and Firefox

Deployment Environment: MS Azure

Operating System: Ubuntu 20.04 Server

Database: MySQL v: XX

Web Server: NGINX XX

Server-Side Language: node.js

Web Framework: Express

IDE: WebStorm, Visual Studio Code

8. Team and roles

Name	Role
Alexander Wiegel	Team Lead
Rakesh Kumar	Backend Developer
Abdullah Butt	Backend Developer
Chahat Soni	Frontend Developer
Maksim Kanushin	Backend and Github Master
Ziad Khaled	Frontend Developer

9. Checklist

Activity	Status (Done/OnTrack)
Team found a time slot to meet (online) outside of the class	Done
GitHub master chosen	Done
Team decided and agreed together on using the listed SW tools and deployment server	Done
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	Done
Team lead ensured that all team members read the final M1 and agree/understand it before submission	Done
GitHub organized as discussed in class (e.g. master branch, development branch, folder for milestone documents)	Done