

# “Master Team Project WiSe 2021/22”

## “MentorMe”

**(Team C)**

## Team Members

- Yogeeta Sharma (yogeeta.sharma@informatik.hs-fulda.de) (Team Lead)
- Ahmad Estaitia
- Ankit Anand (GitHub Master)
- Bibek Gaihre (Backend Lead)
- Mohit Dalal
- Mohammed Afwan
- Omar Ibrahim
- Pratikkumar Arvindbhai Kakadiya (Frontend Lead)

## "Milestone 1"

**Date: 23.11.2021**

## History Table

[illegible]

## Executive Summary



MentorMe is an online platform with a vision of redefining the tutoring ecosystem. This is the last application any student or a tutor would sign up to. It aims to connect the best in the domain tutors & SMEs to students. The application is powered by cutting-edge tech stack and is completely cloud based.

Germany itself is a very big education hub with over 300k international student enrolled currently as per the Daad website. The students are always in hunt of assistance for their subjects' post the lectures and their semester break and with no dedicated platform most of them fail to find a good subject/domain coach. MentorMe is a one-stop solution for their needs.

The easy 3 step on boarding process makes it hassle-free for the tutors to sign up into the application. The thorough profile verification of the tutors by a backend team ensures we only have authentic and trusted tutor profile. Once a student signs up, he/she can book a tutor and have 3 free trial classes initially. The lectures could also be recorded and watched offline from the app for 2 times. Further watch request would come with some nominal charge. The lectures are hourly charged and the payment by the tutors are made to the platform. After a nominal service fee deduction the tutors are paid every month end. Our tutor review system ensures that the students get to review the tutors based on their merit so that the good ones stand out in the application. In the long term, we aim to partner with universities which don't have an online-ecosystem and struggle once their lectures go online due to situation like the current pandemic. These universities could use our platform to host online lectures and archive them too.

Our team is composed of 8 members with very diverse technological experience. We have an amalgamation of very skilled backend, frontend, devOps and project management engineers who have proven their mettle in the industry with big tech organizations. The potential of the market and our vision drives us to build this fool-proof robust EdTech product.

## **Personae**

### **Tutor**

A tutor, often known as an academic tutor, is someone who assists or mentors one or more persons in certain subject areas or abilities. The tutor's purpose is to spend a few hours each day, week, or month passing on their knowledge of the topic or skill to the students. The tutor may make assignments, initiate a conversation, submit questions to a forum, construct lessons, and the tutor can design and deliver a quiz, and you can keep track of all of these actions, and many more, using a range of administrative tools. A tutor's pain point is what they are meant to tolerate and overlook, the most significant of which is the vastly varying ability levels of the students that attend your class. The Tutor can't reject the student when he/she accepts the student.

### **Student**

A student is a person who is enrolled in an educational institution and who is pursuing knowledge, developing professions, and obtaining work in a selected field. Students may access all of their registered classes at a glance; work on assignments from any computer with an Internet connection and a suitable browser; check grades for assignments, quizzes, and other chores; and check the course calendar for crucial dates and forthcoming course events.

In the application, a user with the Student role can take part in course activities and see resources, but not change them or access the class grade book. If the Tutor has agreed, they will be able to see their own grades. Students can't chat with the tutors, students can't have classes with tutors on our platform. He has to rely on some other platform for live communication and some other platform for classes, on his own arrangement where both tutor and student agree on.

### **Moderator**

The Moderator has special permissions to approve/reject requests of the application. The Moderator may change a student/tutor's profile on request. Many features of a course may be changed via settings by a Moderator (Take off a student from a course). Moderator pain point is that he/she has too many requests to accept or reject.

## **Main Use Cases**

- **Review the Tutor**

A tutor can be reviewed by a student on the basis of how much they liked the Tutor. Ranking from 1-5 in which 1 means that the Student does not recommend that particular Tutor at all and hence didn't like the way they teach and on the other hand, 5 means that the student adores the way of teaching of a particular tutor and hence recommends everyone to learn from that Tutor.

- **Searching a Tutor and subject**

In this application, a student can look or search for a particular subject or for a particular teacher. They'll have the option to filter out the results of the courses on the basis of Teacher and Subject both or any one of these, as they prefer.

- **Approve/Disapprove tutor's content**

A Moderator is someone who can control the content that is put up by a tutor for the students. This means they have authorization to approve or disapprove of the content that is provided on the platform by the Tutor.

- **File Upload**

A Tutor is in the position where he/she has to initially upload his/her CV. And later their course content too.

- **On boarding of a tutor**

Tutor will be on-boarded by the Moderator.

## **List of main data items and entities**

- **Authentication**

Authentication will be used by all the users to access the data of the website

- **Uploading**

Tutor needs to upload their CV and profile picture

- **Searching**

Searching for a tutor of a specific subject or name based can be done on this website

- **Approve/Reject**

The Moderator of the website either can approve or reject the course content (images) which are being uploaded over the website

- **Available / Unavailable**

A student can connect with the Tutor over his available time. The request can be accepted or rejected

- **Student**

1. On-Boarding on the website: Students need to sign up first to access the data over the website.
2. Tutor Review: A student can give the review of a Tutor on a scale of 1-5
3. Searching a Tutor by Subject or Name: A student can search the Tutor of relevant subject
4. Book a Tutor: A student can request for appointing the tutor

- **Moderator**

1. Approve/Reject Tutor Content: Moderator can approve or reject the profile on the basis of CV or profile picture of a tutor
2. Moderator Privileges: Moderator can monitor all the data over the website

- **Tutor**

1. On-Boarding on the website: Tutor needs to sign up first and upload the relevant content
2. Uploading Relevant Content like a profile picture and CV: Tutor needs to upload his/her appropriate CV or Profile picture
3. Accepting the student request: If any student wants to learn from a specific tutor, he/she will get the request which the tutor either can approve or reject.

## **Initial list of functional requirements**

### **Students**

- System must allow new students to register in the system using university email.
- Students must log in using the university email in the system.
- The system must offer students a view and to search for tutors on the system.
- Students must be able to contact the available tutors via real time messaging system.
- Students must be able to enrol in the courses made available by tutors.
- Student must be able to unroll the courses if he/she is not satisfied based on terms and conditions set.
- Students must be able to review the course after it is completed
- Students must be able to change password
- Students must be able to use forgot password functionality.

### **Tutor**

- The tutor must be able to register in the system and use it without any problem once the verification is completed by the Moderator.
- The tutor must be able to log in to the system.
- The tutor shall be able to upload required information (CV, comments, images, video, audio, etc.)
- The tutor shall be able to set pricing for his Tutorial Product.
- The tutor shall be able to edit and update his/her uploaded contents.
- The tutor must be able to view the reviews made by students to his/her course.
- The tutor must be able to change password.
- The tutor must be able to use functionality of forgot Password.

## **Moderator**

- The system must notify the Moderator when new Tutor information arrives.
- Moderator should be able to approve/reject content posted by the tutor before making it live.
- Moderator must be able to ban a user from using the system if he/she is suspicious or harmful to the system.
- Moderator must be able to view the list of all the users, tutors, and classes along with their details.
- The Moderator must be able to change password.
- The Moderator must be able to use functionality of forgot Password.

## **List of non-functional requirements**

- 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 chosen cloud server
- Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers
- All or selected application functions must render well on mobile devices
- Data shall be stored in the database on the team's deployment cloud server
- No more than 50 concurrent users shall be accessing the application at any time
- Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users
- The language used shall be English (no localization needed)
- Application shall be very easy to use and intuitive
- Application should follow established architecture patterns
- Application code and its repository shall be easy to inspect and maintain
- Google analytics shall be used (optional for Fulda teams)
- No email clients shall be allowed.
- Pay functionality, if any (e.g., paying for goods and services) shall not be implemented nor simulated in UI.
- Site security: basic best practices shall be applied (as covered in the class) for main data items

- Application shall be media-rich (images, video, etc.). Media formats shall be standard as used in the market today
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
- 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
- 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).

### **Competitive analysis**

Features	MentorMe	Tutorme	ClubZ	Etutorworld
Tutor Reviewing	✓	✓	✓	✓
In-app Payments	✗	✓	✓	✓
Searching for Tutors based on subject	✓	✓	✓	✓
In-app messaging	✗	✓	✓	✓
In-app online tutoring sessions	✗	✓	✓	✓



Our competitive advantage is that the tutors that register on our application can add their own rate per hour based on the services they provide and their expertise. While in the competitive products, there is a fixed rate that should be paid regardless of the tutor. So our application can be more cost-effective and the users will not have to pay for things they do not need just because there is a fixed rate that has to be paid.

## **High-level system architecture and technologies used**

### **Technology Stack**

- React - 17.0.2
- (Bootstrap 4, CSS 3)
- Node.js – 16
- Express - 4.17.1
- MySQL - V8
- NGINX - 1.17.0

### **Supported Browsers**

- Chrome 60.00 and above
- Firefox 60.00 and above
- Safari 12.00 and above

### **Deployment platform**

- AZURE

### **IDE**

- VS Code – 1.62.3

## Team and roles

- Afwan (Team Member Frontend)
- Ahmed (Team Member Backend)
- Ankit (GitHub Master)
- Bibek (Backend Lead)
- Mohit (Team Member Frontend)
- Omar (Team Member Backend)
- Pratik (Frontend Lead)
- Yogeeta (Team Lead)(Document Master)

## Checklist

- Team found a time slot to meet (online) outside of the class (**On Track**)
- 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 (**On Track**)
- Team lead ensured that all team members read the final M1 and agreed/understand it before submission (**On Track**)
- GitHub organized as discussed in class (e.g. master branch, development branch, a folder for milestone documents, etc.) (**Done**)