

Tutorly Hackathon Project Documentation

1. Project Overview

Project Name: Tutorly

Project Goal: Tutorly is a gamified online tutoring platform for university students that makes learning engaging, personalized, and affordable. It allows students to book sessions, track progress, and interact with a virtual pet system to encourage consistent learning.

Target Users / Stakeholders: University students seeking affordable, flexible, and engaging tutoring.

Deployment: <https://bluesoul2003.github.io/Ctrl-Z-Us/>

2. Problem Statement

The main problems Tutorly addresses:

- **Expensive** - High cost limits accessibility for students
- **Poor Engagement & Lack of Motivation** - Traditional tutoring can feel routine and boring
- **Lack of Personalization** - One-size-fits-all tutoring approaches don't meet individual needs
- **Rigid Schedule** - Students struggle to fit sessions into busy university timetables

3. Our Solution

Affordable

- Price range from RM10 - 35 per hour
- Multiple pricing tiers to accommodate different budgets

AI Integrated

- Recommend the most suitable tutor based on student preferences
- Auto matches schedules to avoid clashes
- Smart matching algorithm for optimal tutor-student pairing

Flexible

- Learn anytime, anywhere
- On-demand or pre-booked sessions
- Choice of online or physical sessions

Gamification

- Virtual pet system to encourage consistent learning
- Earn points, badges, and rewards by learning
- Progress visualization to maintain motivation

Progress Tracking

- Track academic growth over time
- Personalized dashboard showing growth metrics
- Visual progress indicators for better understanding

4. Tech Stack

Frontend

- React
- Tailwind CSS

Backend

- FastAPI
- Gemini API
- WebRTC (for real-time communication)

Database

- Firebase
- Firestore (for real-time data)

5. Development Timeline (Planned Full Implementation)

This timeline represents our complete 8-phase implementation plan for the full Tutorly platform. Currently, only Phase 1 (Frontend UI) has been completed as a working prototype during the hackathon period.

1. **Landing Page + UI (React + Tailwind)** - Phase 1 Completed
2. **Auth System (Firebase Auth)** - Phase 2 - Planned
3. **User Database (Firestore)** - Phase 3 - Planned
4. **AI Assistant Chatbot (Gemini API + FastAPI)** - Phase 4 - Planned
5. **Booking System (Firestore)** - Phase 5 - Planned
6. **Tutor Matching (Firestore queries)** - Phase 6 - Planned

7. **Notification System (Firebase)** - Phase 7 - Planned

8. **Deployment** - Phase 8 - Planned

6. Features / Functionality

Feature	Description
User Sign Up / Login	Students and tutors can register and log in via Firebase Auth
Dashboard	Personalized dashboard showing progress and earned rewards
Session Booking	Students can book tutoring sessions with flexible time slots
AI Tutor Matching	AI recommends tutors based on student preferences and learning style
AI Assistant Chatbot	Integrated chatbot using Gemini API for learning support
Learning Modules	Students can select modules that fit their academic needs
Gamification System	Virtual pet system with points, badges, and rewards
Affordable Pricing	Hourly rates from RM10 to RM35
Progress Tracking	Track academic growth with visual progress indicators
Responsive Design	Supports desktop and mobile views
Real-time Communication	WebRTC integration for live tutoring sessions
Notification System	Firebase-based notifications for session reminders

7. Technical Architecture

Current Implementation:

- **Frontend:** React with component-based architecture
- **Build Tool:** Vite for fast development and hot reload
- **Styling:** Tailwind CSS for responsive design
- **Data:** Mock data for demonstration purposes
- **Deployment:** GitHub Pages for frontend demo

Planned Integration:

- **Backend API:** FastAPI for AI integration and data processing
- **Database:** Firebase/Firestore for real-time data synchronization
- **AI Integration:** Gemini API for chatbot and matching algorithms
- **Video Communication:** WebRTC for live video tutoring sessions between teachers and students

8. Development Process (Frontend Prototype Implementation)

Planning & Design

- Task division among team members for frontend components
- UI/UX design planning for student and tutor interfaces
- Technology stack selection (React + Tailwind CSS)

Prototype Development

- Developed individual React components for each major feature
- Created comprehensive frontend UI including:
 - Student and tutor dashboards
 - Session booking interface
 - Messaging system with group discussions and private chats
 - Learning modules and progress tracking
 - Virtual pet gamification system
 - Tutor search and matching interface
- Implemented responsive design for mobile and desktop compatibility

Accessibility Implementation (Partial)

- Some components include accessibility features (messaging system with ARIA labels, keyboard navigation)
- Remaining components currently lack comprehensive accessibility features
- Future development will complete OKU-friendly features across all components

AI-Assisted Development

- ChatGPT used for generating component templates and logic suggestions
- Claude used for code review and optimization

- All AI-generated code was reviewed and customized before integration

Testing & Integration

- Component integration and bug fixing
- Responsive design testing across different devices
- Mock data implementation for demonstration purposes

Deployment

- Frontend prototype deployed on GitHub Pages for demonstration

9. Challenges & Solutions

Challenge	Solution
Real-time communication implementation	Integrated WebRTC for peer-to-peer video sessions
AI integration complexity	Used FastAPI as middleware between frontend and Gemini API
Database synchronization	Implemented Firestore for real-time data updates
Multi-platform responsiveness	Utilized Tailwind CSS for consistent responsive design
Authentication security	Leveraged Firebase Auth for secure user management
Scalable architecture	Designed modular components for easy maintenance and updates

10. Future Improvements

Advanced Features

- Accessibility upgrades for OKU: text-to-speech, real-time captions, dyslexia-friendly mode
- Group study rooms & peer-to-peer learning functionality
- Multilingual support (BM, English, Mandarin, Tamil) for broader accessibility

Partnership

- Collaborate with schools, NGOs, and government bodies to reach underserved communities
- Corporate partnerships (scholarships, sponsored learning for B40 families)

- Integration with educational institutions and learning management systems

Platform Growth

- Expand beyond Malaysia to other Southeast Asian countries
- Support more curricula and educational levels (primary, secondary, tertiary)
- Mobile app development for iOS and Android platforms

11. References

AI Assistance:

- ChatGPT for component generation and logic suggestions
- Claude for code review

Video References:

- YouTube: "Building Accessible Web Applications" (2024):
<https://youtu.be/hn1lkJk24ow?si=a8UjmwHwuT9DmXvL>
- YouTube: "React Tutorial for Beginners" (2024):
<https://youtu.be/SqcY0GIETPk?si=XmkXWcqW3lhc4sxD>

Research Papers:

- Gamification of e-learning in higher education: a systematic literature review, Smart Learning Environments (2023):
<https://slejournal.springeropen.com/articles/10.1186/s40561-023-00227-z>
- ScienceDirect (2024):
<https://www.scienceDirect.com/science/article/pii/S0047272724000185>
- Investigating the impact of gamification components on online learners' engagement, Smart Learning Environments (2024):
<https://slejournal.springeropen.com/articles/10.1186/s40561-024-00336-3>
- Tutoring in (online) higher education: Experimental evidence, ScienceDirect (2022):
<https://www.sciencedirect.com/science/article/abs/pii/S0272775722001236>

Technical Documentation:

- React Documentation: <https://reactjs.org/docs/>
- Tailwind CSS Documentation: <https://tailwindcss.com/docs>
- Firebase Documentation: <https://firebase.google.com/docs>
- FastAPI Documentation: <https://fastapi.tiangolo.com/>

Deployment Resources:

- GitHub Pages Deployment Guide: <https://docs.github.com/en/pages>