

Project Proposal

Title: CampusGuardian (JU Wisdom – Mentorship & Knowledge Hub)

Course title: Mobile Application Development

Course code: CSE-410

4th Year 1st Semester Examination 2024

Submission Date :30/08/25



Submitted to-

Dr. Md. Ezharul Islam

Professor

Department of Computer Science and Engineering

Jahangirnagar University

Samsun Nahar Khandakar

Assistant Professor

Department of Computer Science and Engineering

Jahangirnagar University

Ahsin Abid

Lead Engineer at Samsung Research

Sl	Class Roll	Exam Roll	Name
01	383	210903	Abdullah Nazmus-Sakib



Project Proposal: JU Wisdom – Mentorship & Knowledge Hub

1. Overview of Project Proposal

JU Wisdom is a digital mentorship and knowledge-sharing hub built as part of the **CampusGuardian** ecosystem. Its goal is to bridge the gap between students, alumni, and faculty mentors of Jahangirnagar University by offering structured mentorship, skill-based exchanges, and AI-powered knowledge access.

The hub empowers students to receive academic, professional, and personal guidance while fostering a culture of **peer-to-peer learning** and **community-driven knowledge sharing**.

2. What Problem I Am Solving

At Jahangirnagar University, students face challenges in accessing mentorship and knowledge resources:

- **✗ No structured mentorship system** to connect students with alumni or retired professors.
- **✗ Knowledge gaps** in career development, skill training, and exam preparation.
- **✗ Over-reliance on informal word-of-mouth guidance**, leading to misinformation.
- **✗ Missed opportunity** to leverage alumni expertise for student empowerment.
- **✗ Limited access to curated academic resources** like FAQs, syllabi, and past experiences.

👉 **JU Wisdom** solves these problems by building a **centralized mentorship and knowledge-sharing hub** with human mentors and AI-driven support.

3. Major Feature List

1. **1:1 Mentorship Sessions**
 - Students can book mentoring slots with alumni or retired professors.
 - Secure audio/video sessions using Agora SDK.
 - Session rating & feedback system for quality assurance.
2. **Micro-Talks (Knowledge Capsules)**
 - Short expert lectures on career, skills, or academics.
 - Stored offline for repeated access.
 - Categorized by themes: Career, Research, Life Skills, etc.
3. **Skill-for-Skill Credits**
 - Peer-to-peer mentorship system where seniors help juniors.

- Earn tokens (“Wisdom Credits”) for offering guidance.
 - Redeem tokens to request help from others.
 - 4. **JU KnowledgeBot**
 - AI-powered chatbot trained on JU syllabus, FAQs, and campus information.
 - Offline support with local dataset for essential queries.
 - Answers academic queries, campus event schedules, and general FAQs.
-

4. Alike App Name and Your Major Novelty

Similar Apps/Platforms:

- **LinkedIn Learning / LinkedIn Mentorship** → Career mentorship, but global and not JU-specific.
- **Coursera / Udemy** → Course-based, but not personalized mentorship.
- **Google Socratic** → AI academic help, but lacks mentorship community.

Major Novelty of JU Wisdom:

- 🎯 **Campus-specific** → Tailored for Jahangirnagar University’s context, syllabi, and alumni.
 - 🔄 **Hybrid Mentorship Model** → Combines **alumni mentoring** + **peer skill-sharing** + AI support.
 - 🎁 **Gamified Mentorship Economy** → Skill-for-skill “Wisdom Credits” encourages sustainable sharing.
 - 📖 **Offline Knowledge Capsules** → Stored talks and AI data for low-connectivity students.
-

5. AI Feature to Integrate

JU KnowledgeBot (AI Mentor Assistant)

- **Tech:** OpenAI API + fine-tuned local LLM fallback.
 - **Functions:**
 - Answer syllabus-related queries (e.g., past exam patterns, course outlines).
 - Provide study tips and academic FAQs.
 - Suggest mentors/micro-talks based on student’s profile.
 - **Offline Mode:** Core knowledge base cached using SQLite/Hive for rural connectivity resilience.
-

6. Basic UX Design (Wireframe-Level Flow)

Home Dashboard (Mentorship Hub)

👉 Four main cards:

1. **Find a Mentor** → Search alumni/professor, book session
2. **Micro-Talks** → Browse/listen to stored sessions
3. **Skill Exchange** → Offer help / redeem credits
4. **KnowledgeBot** → Chat interface

Session Booking Flow:

- Select mentor → View availability → Book slot → Confirm → Join session (video/audio)

Skill-for-Skill Flow:

- Post request (“Need help with dart Programming”) → Match with senior → Earn/spend tokens

KnowledgeBot Flow:

- Chat UI with text + voice → Suggested links/resources → “Ask Mentor” option

7. High Level Design (Folder Structure & Class Names)

Project Folder Structure (Flutter)

```
lib/
├── main.dart                # Entry point
├── config/                  # App-wide configuration
│   ├── app_theme.dart      # Colors, typography
│   ├── app_routes.dart     # Route management
│   └── constants.dart      # String keys, tokens
├── core/                    # Reusable core logic
│   ├── services/           # External services integrations
│   │   ├── firebase_service.dart
│   │   ├── agora_service.dart
│   │   ├── ai_bot_service.dart
│   │   └── storage_service.dart
│   ├── utils/              # Helpers, formatters
│   │   ├── validators.dart
│   │   └── date_utils.dart
│   └── widgets/            # Shared UI components
│       ├── app_button.dart
│       ├── app_textfield.dart
│       └── loading_spinner.dart
├── features/                # Each feature = self-contained module
│   ├── mentorship/         # 1:1 mentorship
│   │   ├── models/
│   │   │   ├── mentor.dart
│   │   │   └── session.dart
│   │   └── screens/
```

```
├── ┌── ┌── mentor_list_screen.dart
│   │   ├── mentor_detail_screen.dart
│   │   ├── session_booking_screen.dart
│   │   └── session_call_screen.dart
│   ├── controllers/
│   │   └── mentorship_controller.dart
│   └── widgets/
│       ├── mentor_card.dart
│       └── booking_form.dart
├── microtalks/ # Micro-talks storage
│   ├── models/
│   │   └── talk.dart
│   ├── screens/
│   │   ├── microtalks_list_screen.dart
│   │   └── talk_detail_screen.dart
│   ├── controllers/
│   │   └── microtalks_controller.dart
│   └── widgets/
│       ├── talk_card.dart
│       └── download_button.dart
├── skill_exchange/ # Peer-to-peer skill sharing
│   ├── models/
│   │   ├── request.dart
│   │   └── credit.dart
│   ├── screens/
│   │   ├── request_list_screen.dart
│   │   └── offer_help_screen.dart
│   ├── controllers/
│   │   └── skill_exchange_controller.dart
│   └── widgets/
│       └── request_card.dart
├── knowledgebot/ # AI-powered chatbot
│   ├── models/
│   │   └── bot_message.dart
│   ├── screens/
│   │   └── chat_screen.dart
│   ├── controllers/
│   │   └── chat_controller.dart
│   └── widgets/
│       ├── chat_bubble.dart
│       └── input_field.dart
└── shared/ # Cross-feature reusable code
    ├── network/ # API / HTTP logic
    │   └── api_client.dart
    └── storage/ # Local storage helpers
        ├── hive_helper.dart
        └── sqLite_helper.dart
```

Key Class Names

Mentorship Module

- Mentor (model)
- Session (model)
- MentorshipController (business logic)
- MentorListScreen, SessionBookingScreen, SessionCallScreen

Micro-Talks Module

- Talk (model)
- MicroTalksController
- MicroTalksListScreen, TalkDetailScreen

Skill Exchange Module

- Request, Credit (models)
- SkillExchangeController
- RequestListScreen, OfferHelpScreen

KnowledgeBot Module

- BotMessage (model)
- ChatController
- ChatScreen, ChatBubble

✅ This is a **Flutter-idiomatic modular structure**:

- Each **feature is self-contained** (models, screens, widgets, controllers).
 - Reusable UI goes to **core/widgets** or **shared/**.
 - Services like **Firebase, AI Bot, Agora** go under **core/services/**.
-