PFW Chatbot PROPOSAL

*Neel Singh, Koustav Mallick, Sanidhya Sharma, Daniyal Parveez*

**Associated course / instructor**: CS 59000 -- Applications of Deep Learning / Dr. Zesheng Chen

**Problem statement**:

The process of thinking about, enrolling into, and working through any program at an educational institution can be a daunting process on multiple fronts. As such, the journey is bound to evoke a range of questions in the hearts and minds of students and parents alike.

In this digital age, these institutions are bound to have comprehensive online resources, and more traditional channels for addressing questions, and providing support and information in general. However, this information can often be disorganized and hard to find. Reaching out to concerned authorities in-person, or via phone, chat or mail can be a slow process.

We believe that this entire process of providing guidance to students, parents and others in need of information can be improved by leveraging modern AI techniques. An intelligent chatbot that has been fine-tuned on data derived from existing university resources should be able to resolve a large number of queries with little to no oversight.

While targeting the larger education ecosystem would be a more entrepreneurial undertaking beyond the scope of our academic project, we believe that we could produce a chatbot that could work well for our own MS CS program.

**Features**:

Our proposed bot will be able to

1. Respond quickly and accurately to queries.
2. Elevate the student experience by providing instant access to information.
3. Direct users to the correct point of contact,
4. Offer up-to-date information based on the approved data provided by the department.

**Benefits**:

Implementing this bot will

1. Enhance student satisfaction through quick, accurate responses.
2. Save staff time and resources.
3. Improve operational efficiency.
4. Enable data collection for service enhancements.

**Implementation**:

Our implementation plan includes

1. Identifying common queries and creating a knowledge base.
2. Selecting and customizing a suitable chatbot platform.
3. Training the chatbot and fine-tuning its accuracy.
4. Integrating it with our website (if possible).
5. Conducting a pilot test phase for user feedback.

**Must-Have Requirements (Functional):**

1. Basic Chat Functionality: The chatbot must be able to engage in natural language conversations with users.
2. User Onboarding: It should provide a seamless onboarding process for users, including a welcoming message and instructions on how to use the chatbot.
3. Information Retrieval: The chatbot must be able to provide information about college programs, admission requirements, course schedules, campus facilities, and other relevant college-related information.
4. Personalization: The chatbot should be able to personalize responses based on user data, such as program of study, academic standing, or preferences.
5. FAQ Handling: It must be capable of handling frequently asked questions about the college, admissions, tuition, and academic policies.
6. Event Notifications: Notify users about important events, deadlines, or campus updates (e.g., registration deadlines, open houses, exams).
7. Integration with College Systems: Integrate with the college's databases and systems to retrieve up-to-date information, such as class schedules or academic announcements.
8. Course Registration Assistance: Provide support for course registration, including checking course availability and helping with registration-related queries.

**Must-Have Requirements (Non-Functional):**

1. Deployment: Website Hosting (Or access to the department website)
2. Data Access and Integration: Read-Only access to the department database and documents (for training and result purposes).
3. Infrastructure: Server (Cloud Service like AWS/Azure/GCP) for backend services.
4. Security: Ensure that user data and interactions are secure and compliant with relevant privacy regulations.
5. Reliability: The chatbot should be available and responsive around the clock, minimizing downtime.
6. Scalability: The system should be scalable to accommodate an increasing number of users, especially during peak times such as registration periods.
7. Usability: The chatbot should have an intuitive and user-friendly interface, making it easy for users to interact with.

**Nice-to-Have Requirements (Functional):**

1. Integration with Brightspace: to provide course-related information.
2. Campus Map and Directions: Assist users in finding locations on campus and provide directions.
3. Feedback Mechanism: Allow users to provide feedback on chatbot interactions to continuously improve its responses.
4. Language Support: Provide multilingual support for international students or users who prefer other languages.

**Nice-to-Have Requirements (Non-Functional):**

1. User Analytics: Implement analytics to track user interactions, popular queries, and user satisfaction to improve the chatbot over time.
2. Customization: Allow users to customize their chatbot experience (e.g., theme, notification preferences).
3. Performance Optimization: Optimize the chatbot's performance to minimize response times and resource consumption.
4. Voice Interaction: If feasible, enable voice-based interactions with the chatbot.
5. Cross-Platform Compatibility: Make the chatbot accessible on various platforms, including web, mobile, and social media.
6. Data Backup and Recovery: Implement robust data backup and recovery procedures to prevent data loss.
7. Accessibility: Ensure compliance with accessibility standards to make the chatbot usable by individuals with disabilities.