My Choice: Python with JavaScript

Reasoning:

- **AI and NLP**: Python is better suited for implementing the AI and NLP components due to its powerful libraries and frameworks.
- **Backend Development**: Python with Flask offers a robust backend environment, especially with easy MongoDB integration through `pymongo`.
- **Web Interface**: JavaScript (React) for the front-end will provide a modern and interactive user experience.
- **Resource Sharing:** I used Flask-Cors to share resources between the backend and frontend.

Architecture & Implementation Details:

1. **Backend (Python)**:
    - Use Flask to create a REST API.
    - For the NLP & AI part I started using `spaCy` or `NLTK` to parse natural language input. However, I ended up using regular expressions with the module `re` to handle this part.
    - Handle database operations with `pymongo`.
2. **Front-End (JavaScript)**:
    - Develop a responsive web page using React.
    - Implement form submission and interaction with the backend API.
3. **Database (MongoDB)**:
    - Set up a MongoDB cluster on MongoDB Atlas.
    - Create a database called tutorDB.
    - Create a collection for tutor profiles called tutors and store availability data.

By leveraging the strengths of both Python and JavaScript, I created a robust and efficient Calendar Co-pilot system that meets all the project requirements.

Areas of Improvement:

Enhanced User Experience: Implement more interactive and visually appealing feedback mechanisms to enhance user experience.

Deployment: Use Heroku for the backend and Vercel for the frontend.

Advanced NLP: The use of a more robust NLP model for more accurate mapping of the tutor's natural language to time slots.