Project Proposal - Schedulr

Group: Donovan Felton, Chris Chatel, Jeron Alford

Verbal Description

This project is a meeting scheduling application designed to help large groups find meeting times that maximize attendance. Coordinating meetings with many participants often leads to friction and excessive back-and-forth communication. This application reduces that friction by collecting participant availability and automatically identifying meeting times that work for the greatest number of people. The project is conceptually similar to tools such as When2Meet, but it emphasizes decision support rather than simple visualization. Many existing tools display availability overlaps and require users to manually interpret results. In contrast, this application analyzes availability data and ranks potential meeting times based on attendance metrics using algorithmic or AI assisted techniques. The goal is to provide clear, data driven recommendations that help users quickly select an effective meeting time.

The workflow begins when a meeting organizer creates an event and selects a range of possible dates and times. Participants receive a unique link and submit their availability through an intuitive web interface. Once responses are collected, the system processes the data and presents a ranked list of meeting options showing how many participants can attend each time slot. If enabled, the application can automatically suggest an optimal meeting time. Optional features may include prioritizing certain participants, enforcing minimum attendance requirements, or exporting the selected meeting time to an external calendar.

The primary users of this application include students, project teams, clubs, study groups, and small organizations that frequently coordinate meetings with multiple participants. The application is designed to be platform independent and accessible through any modern web browser, requiring no installation for end users.

From a technical perspective, the project will be developed as a full stack web application. The front end will handle user interaction and availability input, while the back end will manage meeting data, user responses, and scheduling logic. The system will use common web technologies such as HTML, CSS, JavaScript, and a modern framework. A RESTful API will connect the front end to a relational or document based database. Development will be managed using Git, with future updates focused on usability and feature expansion.

Justification

This project is appropriate for a senior level computer science project course because it brings together concepts we have learned throughout the curriculum at LMU. Building a full stack web application allows us to apply skills such as system design, data modeling, client server communication, and user interface development. These skills directly connect to coursework we have completed in classes like Web Applications, Algorithms, and Artificial Intelligence. Writing the scheduling logic also reinforces algorithmic thinking and breaking down problems into manageable parts.

The project gives our team the opportunity to go beyond basic implementations by dealing with real world concerns such as usability and data consistency. Designing an algorithm that evaluates availability across many participants requires thoughtful choices about data structures and performance. Adding AI based features allows us to explore how automated decision making can be used in a practical and realistic way, rather than in a purely theoretical setting.

The scope of the project is reasonable for a single semester. We can build a minimum viable version early on and then add features gradually as time allows. This makes the project manageable while still giving us room to take on meaningful technical challenges.

Most of the tools and skills needed to complete the project are already within our experience, including programming, web development, and version control. At the same time, the project pushes us to learn new techniques, especially in areas related to AI and system design. No specialized hardware is required, and the entire project can be developed using standard tools. Overall, this project is practical, relevant, and motivating for our team. It addresses a common problem that we have all experienced and gives us the chance to build something useful while applying what we have learned throughout the computer science program at LMU.