Queen's Course Scheduler - Software Requirements Specification

1. Overview
This document outlines the requirements and core functionality for the Queen's Course Scheduler Application, a web-based tool designed to help students generate valid course schedules for the Fall or Winter academic terms.

2. User Functionality

2.1 Course Selection

- **Course Count**:
    - Users must select the number of courses (between 2 and 7 inclusive).
    - Implemented via dropdown menu (default: 5).
- **Term Selection**:
    - Users select the term (Fall or Winter).
    - Used for labeling/exporting schedules (e.g., in email subject lines).

2.2 Course Input Form
Users input each course via a form with the following fields:

- **Course Code** (mandatory)
- **Start Time** (mandatory)
- **End Time** (mandatory)
- **Section** (optional)
- **Lecture/Tutorial/Lab** (optional)
- **Instructor Name** (optional)
- **Building/Room** (optional)

Optional Feature: Mark Course as Mandatory

- Allows the user to indicate that a course *must* appear in every generated schedule.
- UI should clearly communicate that this could reduce the number of generated schedules due to potential conflicts.

2.3 Course List Management

- Submitted courses appear in a list below the form.
- Each course entry supports **Edit** and **Delete** functionality.
- Users can continue adding until the selected number of courses is met.

2.4 Schedule Generation

- Once the course count is reached:
    - A "Generate Schedules" button appears.
    - Clicking it sends the course list to the backend for processing.

2.5 Viewing and Saving Schedules

- Schedules are displayed in a grid or calendar layout.
- Users can mark favorites and **send selected schedules to an email address**.

3. Backend Requirements

3.1 API

- Expose a **POST** endpoint (e.g., `/api/schedule`) to receive course input.
- Responsibilities:
    - Parse course input.
    - Generate all valid combinations.
    - Filter out schedules with time conflicts.
    - Return a JSON list of valid schedules.

3.2 Analytics

Option 1: Custom Event Tracking

- Log "Schedule Generated" events via endpoint: `/api/analytics/schedule-generated`
- Data to log:
    - Timestamp
    - Possibly course count or term

Pros:

- Measures actual usage
- Enables display metrics like "500+ schedules generated"

Cons:

- Doesn't distinguish unique users unless combined with cookies or fingerprinting

Option 2: Lightweight Analytics Tool Integration

Candidates:

- Plausible Analytics
- Fathom

- Umami
- GoatCounter
- (Optional) Google Analytics *(more invasive)*

Pros:
- Easy to integrate, privacy-friendly
- Track sessions, page views, referrers, and geo data

Cons:
- Limited custom tracking unless event logging is configured

4. Future Considerations
- Authentication for saving multiple favorite schedules.
- Optional integration with Queen's course database/API for autocomplete.
- Export schedules as PDF or ICS calendar files.
- Add dark mode for accessibility.