Event Trigger and Email Notification System

Objective:

Your task is to create a simple web-based application using Ruby on Rails (RoR) that integrates with the iterable.com API. The application will have two buttons, each responsible for creating a different event associated with a user. After the events have been created, the application should search through them, and for every Event B, an email notification should be sent via iterable.com API.

Requirements:

- User Interface:
  - Develop a UI using RoR with two buttons:
  - Create Event A: When clicked, creates an instance of Event A associated with a user in iterable.com.
  - Create Event B: When clicked, creates an instance of Event B associated with a user in iterable.com.

- iterable.com Integration:
  - Integrate the application with iterable.com API to create events and manage users.
  - Use the iterable.com API to send email notifications for Event B.
  - Mocking iterable.com
    - API: Ensure that the interaction with the iterable.com
    - API is mocked out using over-the-wire mocks for testing purposes

- Code Structure:
  - Your code should be modular, readable, and maintainable in Ruby on Rails.
  - Include appropriate comments and documentation.

- Bonus:
  - Implement user management within the application, allowing for multiple users to be associated with events.
  - Implement authentication for users using Devise or any other suitable gem.

Constraints:

  - The application must be developed using Ruby on Rails.
  - Follow best practices for integrating third-party APIs in a RoR application and for mocking out API interactions.
  - The application does not need to persist data across restarts, apart from what is stored in iterable.com.

Submission:

- Include all the source code, assets, and any additional documentation.
- Provide instructions on how to set up and run the application.
- Ensure the submission includes over-the-wire mocks for interactions with the iterable.com API.
- Discuss any trade-offs or decisions you made while implementing the solution.

Evaluation:

- Your implementation will be evaluated based on the following criteria:
  - Functionality: Does the application meet the requirements and function correctly?
  - Code Quality: Is the code well-structured, readable, and maintainable within the RoR framework?
  - API Integration and Mocking: Is the iterable.com API integrated correctly, are the interactions appropriately mocked, and does it function as expected?
  - User Experience: Is the application easy to use and understand?
  - Scalability: How well could this solution scale with additional features or increased load?
  - Bonus: Implementation of additional features like user management and authentication.

Hints:

- Familiarize yourself with the iterable.com API documentation for creating events and sending emails.
- Look into suitable libraries or frameworks in Ruby for creating over-the-wire mocks.
- Consider edge cases, such as what should happen if there is an attempt to send an email for Event B but there is no associated
- There is no API key available for Iterable. You will have to mock it