2022 Web Application Group Project.

1. **Aims:**
   1. Understand principles of designing user friendly websites.
   2. Write standards compliant websites in HTML & CSS.
   3. Develop dynamic, client-side web content using Javascript.
   4. Understand the interactions between the client-side and server-side components of web applications.
   5. Design and build AJAX calls to a server and handle responses.
   6. Use third party APIs in web applications.
   7. Understand the role of databases in web applications.
   8. Plan & integrate a database into a web application.
   9. Understand and mitigate security issues faced by web applications.
2. **Project overview:**
   1. Description:

Your group assignment for the Semester is to design and develop a social Web Application that allows users plan events/gatherings and compare calendars to find a suitable time/date for that gathering.

Detailed information on features and functionality is provided in the Requirements and following sections.

* 1. Submission:
     1. This is a big project and can’t be completed at the last minute so to keep you on track, we’ve set two milestones:
        1. A milestone submission at the end of week 9
        2. A final submission at the end week 13.

1. **Core requirements:**
   1. Software requirements:
      1. The web application must be **developed in *and* run in** the VSCode for CS50 IDE.
         1. You should also make regular backups/downloads of your work.
         2. You are encouraged to setup a dedicated GitHub and CS50 account for groupwork with shared access to all group members to avoid accidentally sharing your personal work.
         3. You must get explicit permission from the course coordinator if you want to use a different environment.
      2. The web application must use NodeJS, Express, and AJAX .
      3. The database must be a SQL database.
      4. Your markup/code must pass W3C Validation and ESLint linting.
      5. You may use other libraries/frameworks not addressed in this course, however:
         1. Remember that you're working in a group; all group members are expected to have contributed to and understand the work developed.
         2. Your implementation must still demonstrate your understanding of the concepts taught in this course.
         3. We may be unable to mark work that deviates too much from the course content.
   2. Group requirements:
      1. This assignment must be worked on in groups of 4.
      2. Groups must be formed before the mid-semester break (end of Week 6).
      3. You must register your group in MyUni.
      4. You may form a group with any other students in this course, but it will be easier to organise your group if they are in the same workshop as you.
      5. Postgraduate Students (enrolled in COMP SCI 7207) will be assigned groups.
   3. Feature requirements:
      1. You must implement a social event planning system.
      2. Your system should contain features as described in the "Feature Details" section below.
2. **Feature details:**

Your social event planning web app should include, and will be assessed on, the following features:

* Users should be able to sign up and log in to the system in order to  
  + Sign up/log in.
  + Manage their user information.
  + Create a new event.
  + Generate a link for people without accounts.
  + Specify their availability for an event.
  + Link their calendar to automatically check their availability.
  + See times when everyone is available for an event.
  + Confirm/finalise an event time.
  + Add the finalised event to their calendar.
* System Admins should be able to log in in order to:
  + Manage their user information.
  + Manage Users.
  + Manage Events.
  + Sign-up other Admins.
* Users should be able to choose to link a social media/email/other account, allowing login via that platform, to make logging in easier.
* Users without accounts should be able to specify their availability for an event provided by a custom link.

One (1) of the following three (3) special features must also be present (2 for PG Students):

* Email notifications
  + Users can request that email notifications be sent to them that include updates when users responed, availability is confirmed, the event is finalised, or the event is cancelled.
  + Users should be able to choose which types of email notifications are sent

~ OR ~

* Link to Social Media  
  + The system can setup the planned event on the user's social media platform.

~ OR ~

* Location Planning  
  + Users can specify their home location on a map.
  + The system shows/suggests event locations that are easy to get to for all attendees

1. Forming a group
   1. For UG students this project is completed in groups of 4.

Milestone 1

**Minh – Research**

1. Rubric
   * “You've found several examples each of style, structure & features that they can build on. Detailed justification/discussion is provided for these examples.”
2. Examples of things to include
   * Examples of other sites/apps and which use similar features and how they have implemented it (ex. Google Calendar API)
   * Also include ideas of how other sites have included more complex features and what resources have they used to do so.
   * What has made other event planning sites successful? What features do people like, how are they simplified and made accessible?

**Corey – Data Plan**

1. Rubric
   1. “A complete data plan is given, detailing all interactions between the client and server, what data will be sent, how it will be stored, and what processing is needed.”
2. Examples of things to include
   * Specify where NodeJS, Express, Vue or AJAX might be used when handling data. Which interactions will take place client side and which take place server-side and why.
   * Note which interactions will lead to information being saved or updated in a sql database.
   * What form will the information take? What does the client need to do to retrieve server information? Which server information will only be accessible to admins and how can they access it?

**Reilly – Database schema**

1. Rubric
   1. “A detailed, valid, Schema diagram is provided, showing all Tables, Attributes and Relationships of all data being stored in the database, including Keys & cardinality. Relationships clearly indicate which attributes are involved. The Schema is in 4th or higher normal form.”
2. Examples of things to include
   1. Aside from the complete schema, also relating back how the schema will be applied to the website.

**Annie– Design + Features + Review**

1. Rubric:
   1. “You've provided a detailed, clearly labelled and dimensioned mockup of the main page(s) of your site. User interactions are clearly broken down, accompanied by further mockups where necessary. Clear justification/discussion is provided for these decisions and reflects the research undertaken where applicable.”
   2. “You've provided, or included in your design, detailed planning/breakdown of how each of the features in your website will function.”
   3. “Each of the usability Heuristics, as well as kinetic and cognitive load have been tested for and reviewed against your design. The design has been updated to reflect this, or where updating was not necessary, justification is provided.”
2. Examples of things to include:
   1. Full design, box models and justification for box models and styling, references for why these decisions were made based on other successful websites/apps.
   2. Emphasis on heuristics and the kinetic and cognitive load, emphasizing user accessibility and ease of use
   3. Begin to outline classes and id’s and how they’ll be used in the css file

Final rubric mark implementation we can all contribute to:

1. “The group is well on track to successfully complete their implementation including all features to a high degree of quality.”

**Minh – Research:**

Our website:

Name: Evently

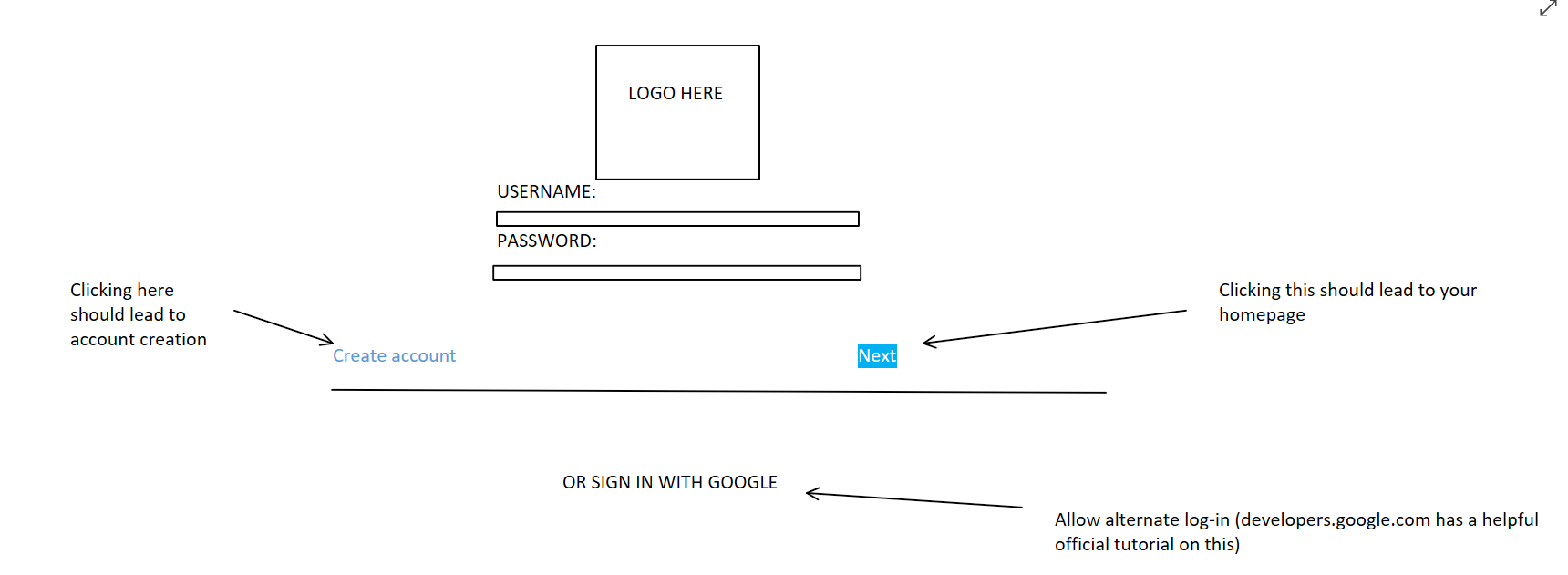
Features:

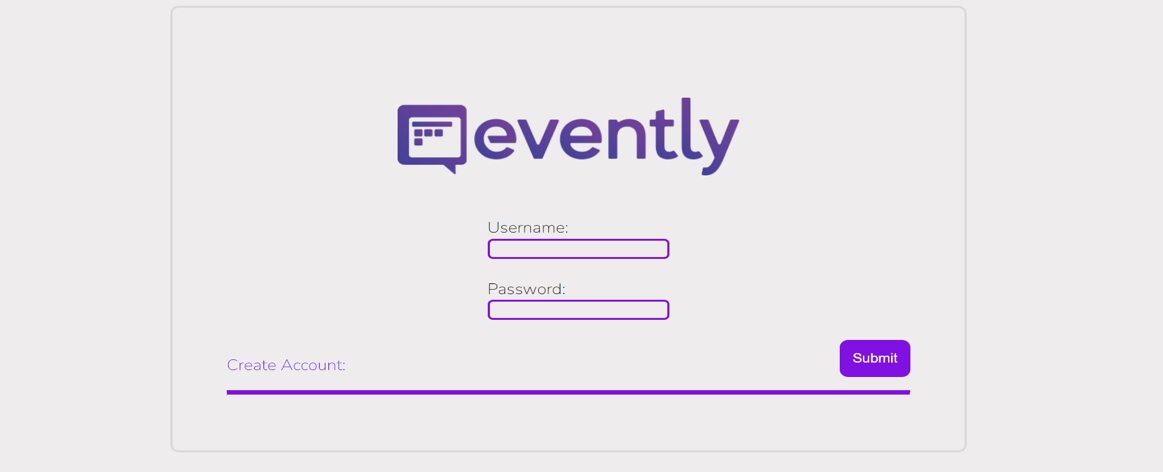
* Sign up/ log in
* Manage the user information
* Search, view calendar events
* Access and edit private calendar events
* Generate a link for people without accounts
* Specify their availability for an event
* Link their calendar to automatically check their availability
* See times when everyone is available for an event
* Confirm/finalise an event time
* Add the finalized event to their calendar.
* Set notifications and reminders
* Send reminders.

Logo: 

Built by Reilly

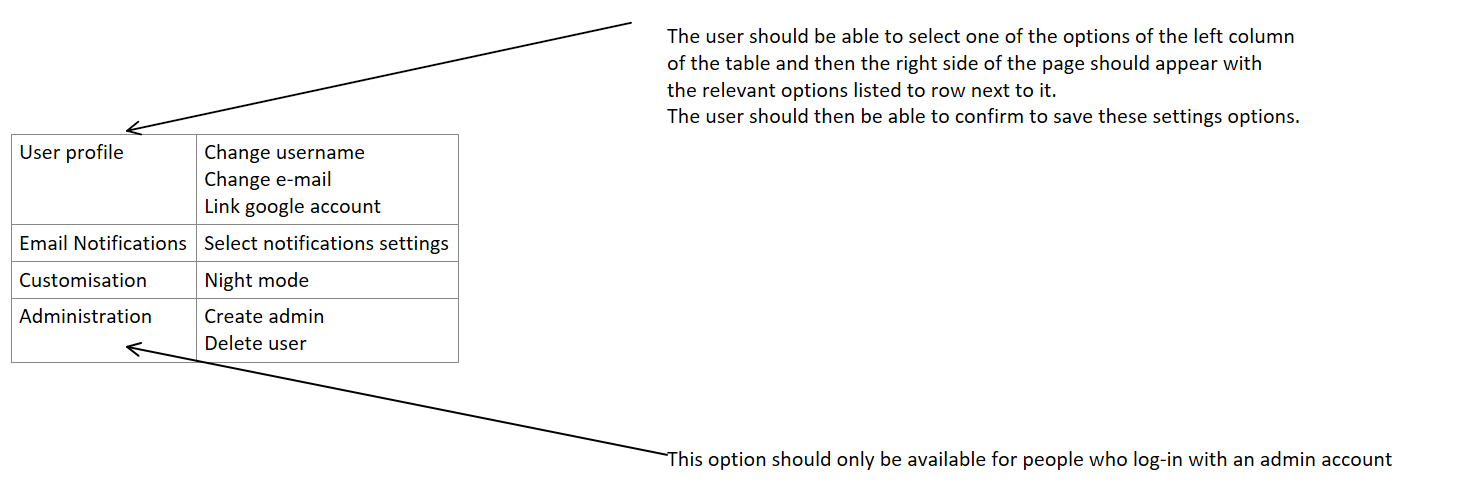
First page:



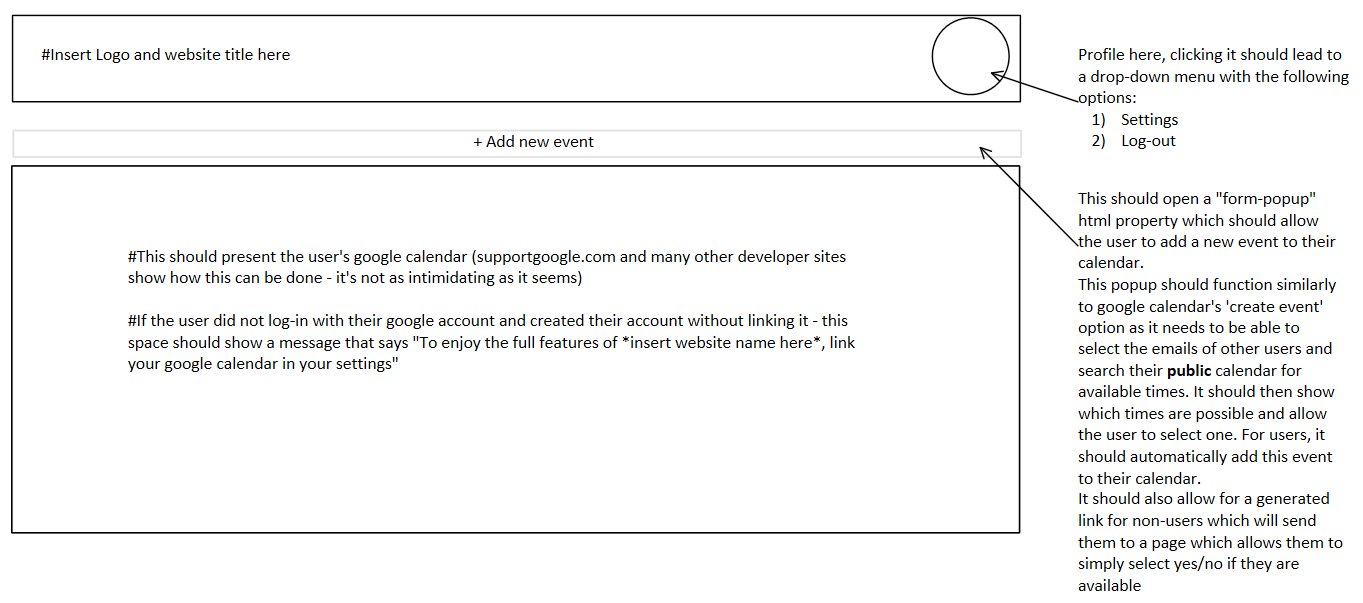


Annie

Setting page:

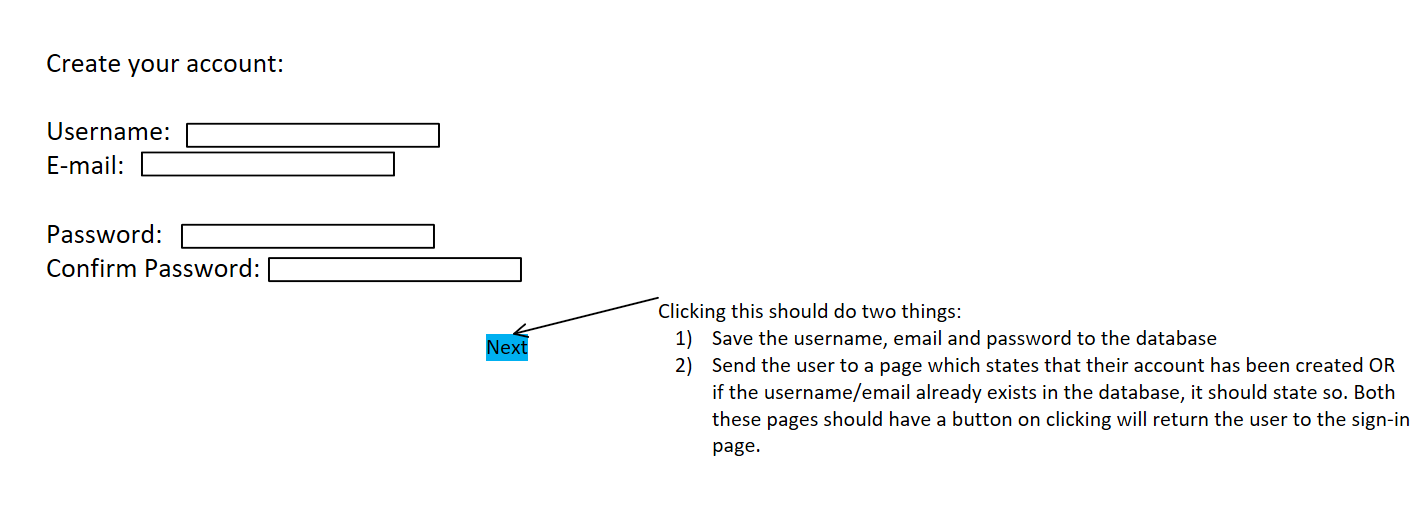
Annie

Calendar page:



Annie

Create account page:



Annie

Examples of websites that can be used for our project:

|  |  |
| --- | --- |
| Name | Features: |
| Google calendar API | Search and view calendar events, access and edit private calendar events, set notifications and reminders |
| Calendarific API | Get public and bank holidays of countries worldwide |
| Super SAAS API | Add online appointment scheduling, send reminders, accept payments, get reports |
| 31 events API | Send invitations to recipients’ calendars, get reports, integrate with other application |
| Nylas Cloud Calendar API | Enable calendar sync capabilities, add scheduling, set reminders, get reports. |

* **Google calendar API:**
  + What is Google Calendar?
  + API Base URL:
    - https://www.googleapis.com/calendar/v3
  + Purpose:
    - Google Calendar helps to share the plan and reminds about it so people can get prepared with no stress.
  + What makes the users want to use Google calendar?
    - The app helps user to organize their schedule
    - The app handles the booking
    - The app adds the event to the calendar
* Users will love the app because they don’t have to manually add those events to their calendars
  + Definition of calendars and events:
    - A calendar is a list of events with additional information such as a summary, default time zone, and location. An ID, which is an email address, is assigned to each calendar. Calendars can be owned by many people.
    - An event is an object that related to something happens at a specific time or time range.
  + What features are in Google calendar API?
    - Types of events:
      * A single event (unique occurrence).
      * A recurring event (multiple occurrences).
    - Organizers
      * Events have a single organizer
      * Events can also have multiple attendees
      * An attendee is usually the primary calendar of an invited user.

Diagram

Description automatically generated

* + - Primary calendars and other calendars:
      * A primary calendar is a form of calendar that is associated with only one user account. This calendar is unique, cannot be changed by other user, and can be shared with other users
      * Other calendars can be modified, deleted and shared among multiple users
    - Calendars and Calendar list:
      * Calendars contain all existing calendars.
      * Calendar list is a collection of all calendar entries that a user has added to their list
    - Time zones
    - Event time zone
  + How do you implement Google calendar?
    - To connect to the Google API, we must first set up the Google API and obtain the various connection details that we will want. Simply activate the calendar API to get started.
      * Go to  [Google API Console](https://console.developers.google.com/apis/dashboard) and follow the steps
    - Search and view calendar events:
      * Add an event:
        + To create an event

calendarID: calendar ID to insert the event into (‘primary’)

event: includes start and end times

Location: place which the event occurs

Attendees: people involve in the event

* + - * + Recurring events
* By creating events, we can search and view calendar events through primary calendars, this is where every user can get up and see which time is suitable for them.
  + - Access and edit private calendar events:
      * Sharing and Attendees
        + Inviting attendees to events

Share an event with others by adding them as attendees.

Sends an invitation email to the attendees and places the event on their calendar

Diagram

Description automatically generated

* + - Set notifications and reminders:
      * Reminders and Notification
        + Get push notification

Set up your receiving URL, or “Webhook” callback receiver

Set up a notification channel for each resource endpoint you want to watch.

* **Calendarific API:**
  + What is Calendarific?
    - Calendarific is a global event intelligence platform and a developer-friendly worldwide holiday API covering public, bank, local, religious holidays and observances for over 230 countries.
  + What are variables?
    - Variables in workflows allow you to store and manipulate data and files. (e.g., Name, Description, Country, Date, etc.)
  + API Based URL:
    - https://calendarific.com/api/v2
  + Purpose:
    - Get public and bank holidays of countries worldwide
    - Set up schedule easier, especially useful for the people that need to work with banks on the worldwide scale.
  + What makes the users want to use Calendarific API:
    - Speed: Over 90% of all requests are served in less than 10 milliseconds thanks to latency-based DNS routing.
    - Reliability: Their auto-scaling cloud servers can handle more than half a billion API requests per day.
    - Up to date: Their data is compiled and confirmed from many sources, and it is updated on a regular basis.
  + What features are in Calendarific API?
    - Get holidays:
      * Holidays will include:
        + Name (Name of the holiday)
        + Description (Description of what’s the holiday is about) -> This will help user to understand about their culture.
        + Type (Type of the holidays, whether it is national, local or religious).
        + Country

ID

Name

* + - * + Date will include:

ISO

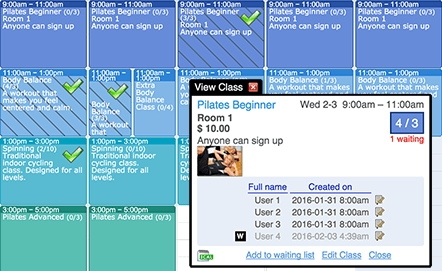
Date-Time

Year

Month

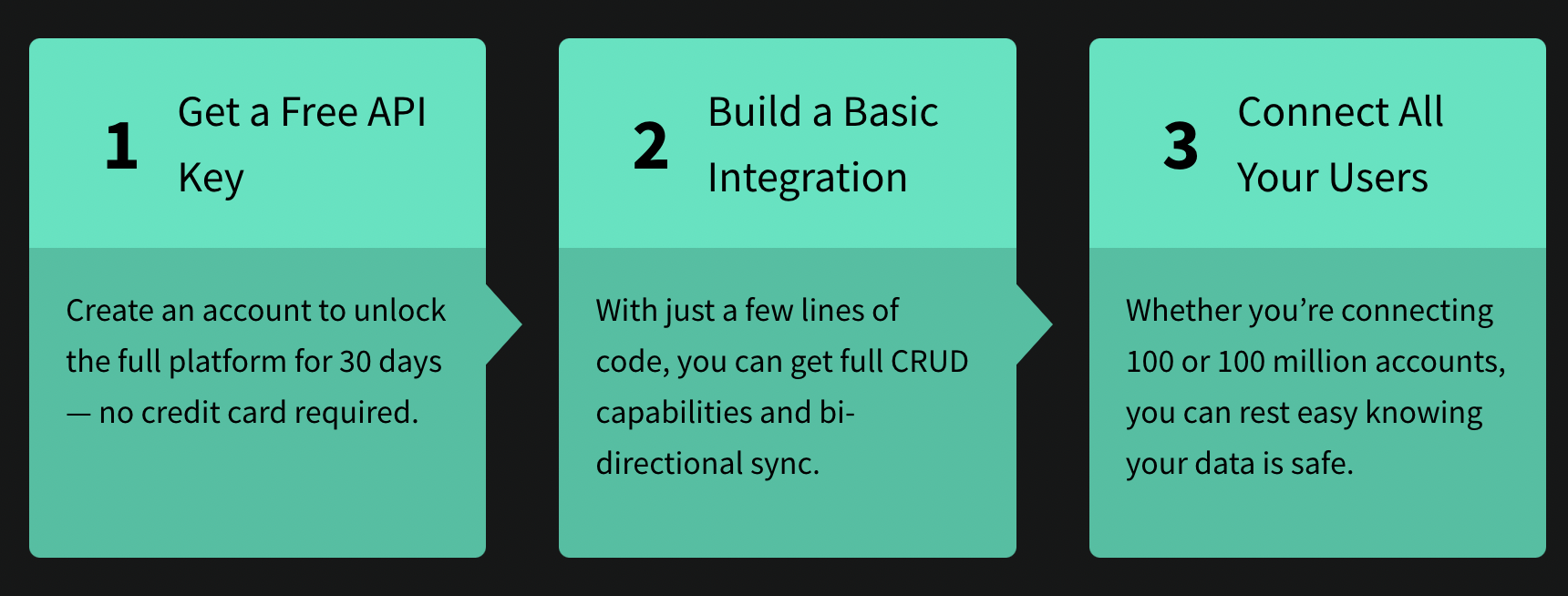
Day

This date-time can be converted to user time zone.

* + How do you implement Calendarific API?
    - Sign up on Calendarific
    - Calendarific will give you an API key
    - Put in the country
      * If we want to implement our country, for example, Australia, then the code part should be:
        + Country: au
    - Implement state code
      * If you don’t give state code, this will give you all the holidays of all the states in that country.
    - Year:
      * The year you want to return the holidays.
* This implementation helps the program to give you the information about the holiday and get it onto your calendar
* **Super SAAS API**
  + What is Super SAAS API?
    - Super SaaS is a powerful online appointment scheduling system that can be integrated into any website or CRM platform.
  + What does Super SaaS API?
    - SAAS stands for Super software as a Service.
  + API URL:
    - https://www.supersaas.com/api/changes/<schedule\_id>.json?from=<last\_retrieval>&api\_key=your\_api\_key
  + Purpose:
    - Get online appointment schedule
    - Create booking system and scale business
    - Set up opening hours
    - Introduce coupons for customers
    - Combine multiple calendars
    - Automated scheduling
  + What makes the users want to use SAAS API:
    - Highly customizable
      * Set up custom design, custom messages, notifications and terms for user
    - Affordable and Scalable
    - Collect payments online
    - Save time and money
      * Send automated confirmations, reminders & follow-ups via SMS or email
    - Combine calendars
    - Share appointment schedule
    - Easy step-by-step appointment scheduling
    - Users can easily book with you in their language  
        
      
    - Send confirmations & reminders:
      * Email & SMS notifications
    - Synchronize with other calendars
    - Security & reliability
  + What features are in Super SaaS API?
    - Client (User):
      * The user’s name or ID or foreign key
    - Schedule\_ID:
      * The ID of the schedule.
    - From:
      * Retrieve appointments starting on or after this time
    - To:
      * Retrieve appointments starting before this time
    - Limit:
      * The maximum number of results you want to return.
    - Api\_key:
      * The administrator API Key for the account the schedule belongs too.
    - Slot
      * Optional, if you let slot = true => additional information will be included
    - Slot\_id
      * Information about the slot this booking belongs to
    - Resource\_ID (Service\_ID)
      * The ID of the resource that was selected
    - Res\_name
      * This is the resource that was chosen if your schedule has more than one resource.
    - Created\_on
      * Creation time
    - Updated\_on
      * Last changed time
    - Waitlisted
      * Check if this booking is booked
  + How do you implement Super SaaS API?
    - First register a Super SaaS account, get your account name and the API key on the [Account Info](https://www.supersaas.com/accounts/edit) page.
    - The client can be used either (1) helper method “instance” or create a new client manually
    - List appointments:
      * This API allows you to retrieve all appointments or slots from a schedule within a time range.
      * Input values:
        + Schedule\_ID
        + From
        + To
        + Today
        + Api\_key
        + Limit
        + Slot
        + User
        + Resource\_ID
      * Output values:
        + ID
        + Res\_name
        + Resource\_ID
        + Slot\_ID
        + Created\_on
        + Updated\_on
        + Waitlisted
    - Agenda:
      * This API allows you to retrieve the appointments of a single user.
      * Input values:
        + Schedule\_ID
        + User
        + From
        + Api\_key
        + Slot
    - To create a new appointment, you need to send an HTTP POST request to /api/bookings.json (or .xml). The request should either contain a JSON or an XML document describing the new user, or have the fields as URI encoded parameters.
* **31 Events API**
  + What is 31 Events API?
    - Is a simple, however, powerful calendaring event management service.
  + API Based URL:
  + Purpose:
    - It enables user to create events and send invitations to the recipients’ calendars, allowing them to easily confirm attendance.
  + What makes the users want to use API:
    - Simple and easy to use
    - Fast, save time
    - Satisfaction because users will have an agreement upon each other.
    - If the customer accepts the calendar invite, they are automatically signed up for the event.
* The design of this calender is simple and easy to use. At the same time, it guarantees to consist all necessary utilities for users.  
  + What does 31 Events API do?
    - Calendar Snack (31 Events) focus on “make it simple”
    - Creating a Calendar Invitation
      * They based on our calendars (user’s calendar) and create the invite “engine”
    - Sending Calendar Invitation:
      * Calendar Snack provides some basic webpage embeds that you may use to convey an invitation to someone. Enter your email address to receive an invitation.Counting Invites Sent & RSVPs  Received (Invitation Responses)
    - Counting Invites Sent & RSVPs  Received (Invitation Responses)
      * It's crucial to make it simple to generate and distribute invitations, but the actual strength of Calendar Snack is the ability to view the numbers.

Diagram

Description automatically generated

* **Nylas Calendar API**
  + What is Nylas?
    - Nylas is a platform that offers a set of APIs for integrating email, calendar, and contacts.
  + API Based URL
  + Purpose:
    - The Nylas Calendar API powers your application with a secure, reliable connection to your user’s calendars. Sync historic and live events into your application in real-time.
    - Allows you to connect to any email service
    - Calendar and event data, such as event titles, locations, descriptions, dates, and more, can be accessed.
  + What makes the users want to use Nylas?
    - Users may access data that is critical to their application's email, calendar, and contacts integration quickly and conveniently.
    - Hundreds of SaaS organizations use Nylas to help them expedite interactions, eliminate context switching from web app to email, and act on valuable email data.
  + What does it offer?
    - Nylas scheduler:
      * Customize front-end scheduling UI.
    - Instant booking
      * Enable users to schedule meetings
    - Route-Robin routing
      * Route meetings evenly between members of a team
      * Time-zone Detection
      * Real-time notifications
      * Service accounts
      * Virtual calendars
      * Event metadata
* As can be seen, Nylas Calendar API provide
  + How does it work?
  + What is a calendar?
    - Each Nylas user can have zero or more calendars, each of which contains a collection of individual events.
  + Types of calendars:
    - Provider
    - Emailed events
    - Virtual calendars (Work just like any other calendar and make it simple to integrate custom scheduling functions into your app.)
  + How do you implement Nylas Calendar API?
    - Get Available Calendars
      * A request to the calendar’s endpoint will return a list of all calendars that the user has access to.
      * Copy the ID of a calendar to which you want to make changes.
      * The is\_primary field is for Google and EWS calendars. If true -> it’s the primary calendar, if false -> secondary or a child calendar.
      * By giving the relevant calendar ID, you can get information about a single calendar.
    - Read Content from Events
      * Make a request to the ‘events’ endpoint to get a list of events from all of a user's calendars.
    - Create, Modify, and Delete Events
    - Modify and Event and Send an Email Invite
      * Notify Participants
    - Get Available Room Resources
    - RSVP to Events
      * Send RSVP statuses to event participants to round out the functionality of your calendar. The send-rsvp endpoint sends an email notification to all of the participants sub object’s email addresses.
    - Delete an event
      * Plans can change. If an event is canceled, it can be erased with a notification issued to all attendees.