```
vue create librarian
Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select, <a> to toggle a
 ◉ Babel
 ◯ TypeScript
 ◉ Progressive Web App (PWA) Support
 ◉ Router
 ◯ Vuex
 ◯ CSS Pre-processors
❯◯ Linter / Formatter
 ◯ Unit Testing
 ◯ E2E Testing
? Choose a version of Vue.js that you want to start the project with
  3.x
❯ 2.x
? Use history mode for router? (Requires proper server setup for index fallback in prod
? Where do you prefer placing config for Babel, ESLint, etc.? In package.json
? Save this as a preset for future projects? (y/N)
```

```
cd librarian
```

Add Vuetify

```
vue add vuetify
found 0 vulnerabilities
✓  Successfully installed plugin: vue-cli-plugin-vuetify

? Choose a preset: (Use arrow keys)
  Vuetify 2 - Configure Vue CLI (advanced)
❯ Vuetify 2 - Vue CLI (recommended)
  Vuetify 2 - Prototype (rapid development)
```

Refer documentation to vuetify v2

```
https://v2.vuetifyjs.com/en/
```

Download logo image

```
https://media.istockphoto.com/id/826736292/vector/librarian-woman-with-books-flat-vecto
```

Defining global styles.css

1. Create new file under `assets` folder `/assets/styles.css`

```css
:root {
    --primary: orange;
    --text-light: #525252;
}

.center {
    position: relative;
    top: 50%;
    transform: translateY(-50%);
    text-align: center;
}
```

1. Import this css file in `App.vue` in styles section as follows

```html
<style>
@import url('@/assets/styles.css');
</style>
```

Define Utils folder

1. Create new folder `utils` under `src`
2. Create 2 files under this folder : `GlobalMixins.vue, AppConstants.js`
3. Define `GlobalMixins.vue` this file as global mixin & `AppConstants` as follows

```js
// main.js
import GlobalMixins from '@/utils/GlobalMixins.vue'
import AppConstants from '@/utils/AppConstants'

Vue.mixin(GlobalMixins)
Vue.prototype.$const = AppConstants
```

```js
//GlobalMixins.vue
<script>
export default {
    methods: {
        randomColor() {
            return "#"+Math.floor(Math.random() * 0x1000000).toString(16)
        }
    },
}
</script>
```

```js
// AppConstnats.js
export default Object.freeze({
    APP_NAME: "Librarian"
})
```

Using CSS icons

Vuetify provides inbuilt support for icons. Read more about this **Vuetify — A Material Design Framework for Vue.js**

Commands to

```
git init
git add .
git commit -m "initial commit"
git branch -M main
git remote add origin https://github.com/mundadakapil/librarian.git
git push -u origin main
```

Command to add existing repo to remote git server

```
git remote add origin https://github.com/mundadakapil/librarian.git
git branch -M main
git push -u origin main
```

Login Module

1. Add Dummy Data
2. Create Login Page
3. Visit Forms section
4. Create Login Structure
5. Create default Object method
6. Create Authenticate method and navigate to books page

Create AppHeader component

1. Design component
2. Define Props

Create Books Module

2. Import list

3. Load data from list

4. Design Book Card & use it.

5. Implement Menu Options

PWA

Install **vue-pwa-asset-generator**

```
npm install --global vue-pwa-asset-generator
```

Generate resource

```
vue-asset-generate -a src/assets/logo.png -o public/img/icons
```

Now got to `vue.config.js` create pwa section and copy following lines

```js
pwa: {
    name: "Librarian App",
    short_name: "Librarian",
    manifestOptions: {
      icons: [
        {
          src: "./img/icons/android-chrome-192x192.png",
          sizes: "192x192",
          type: "image/png",
        },
        {
          src: "./img/icons/android-chrome-512x512.png",
          sizes: "512x512",
          type: "image/png",
        },
        {
          src: "./img/icons/android-chrome-maskable-192x192.png",
```

```
      purpose: "maskable",
    },
    {
      src: "./img/icons/android-chrome-maskable-512x512.png",
      sizes: "512x512",
      type: "image/png",
      purpose: "maskable",
    },
    {
      src: "./img/icons/apple-touch-icon-60x60.png",
      sizes: "60x60",
      type: "image/png",
    },
    {
      src: "./img/icons/apple-touch-icon-76x76.png",
      sizes: "76x76",
      type: "image/png",
    },
    {
      src: "./img/icons/apple-touch-icon-120x120.png",
      sizes: "120x120",
      type: "image/png",
    },
    {
      src: "./img/icons/apple-touch-icon-152x152.png",
      sizes: "152x152",
      type: "image/png",
    },
    {
      src: "./img/icons/apple-touch-icon-180x180.png",
      sizes: "180x180",
      type: "image/png",
    },
    {
      src: "./img/icons/apple-touch-icon.png",
      sizes: "180x180",
      type: "image/png",
    },
    {
      src: "./img/icons/favicon-16x16.png",
      sizes: "16x16",
```

```
        {
          src: "./img/icons/favicon-32x32.png",
          sizes: "32x32",
          type: "image/png",
        },
        {
          src: "./img/icons/msapplication-icon-144x144.png",
          sizes: "144x144",
          type: "image/png",
        },
        {
          src: "./img/icons/mstile-150x150.png",
          sizes: "150x150",
          type: "image/png",
        },
      ],

      start_url: "/",
      display: "standalone",
      background_color: "#000000"
    },
  }
```

Start server

Now you can see new logos visible in Application >> Manifest section

> In development version service worker won't get registered.
> So you will get `No service worker found` error in Application >> Manifest option.

Not recommended

Just to register service worker in development mode

`Go to registerServiceWorker.js file and comment following`

```
if (process.env.NODE_ENV === 'production') {
```