HTML:

1. Создаем папку, в ней создаем файл с расширениями html, css для стилей и js для скрипта.
2. В html в head подключаем фреймворк vue:

<script src="https://cdn.jsdelivr.net/npm/vue@2.7.8/dist/vue.js"></script>

<script src="//unpkg.com/babel-polyfill@latest/dist/polyfill.min.js"></script>

<script src="//unpkg.com/bootstrap-vue@latest/dist/bootstrap-vue.js"></script>

и стили <link rel="stylesheet" href="style.css" />

<link rel="preconnect" href="https://fonts.googleapis.com" />

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<link href="https://fonts.googleapis.com/css2?family=Nunito:wght@300;400;500;600;700;800;900&display=swap" rel="stylesheet" />

1. Подключаем библиотеку chart js для работы с графиками.
2. В теге div cвязываем DOM с данными экземпляра vue: <div id="app".
3. В теге ul с классом mmenuu прописываем подменю в отдельном классе с помощью атрибута href.
4. В отдельном классе через тег img и атрибут src добавляем иконки.
5. Когда происходит нажатие на ссылку, вызываем метод makeActive, определенный как js-сущность Vue.
6. Создаем блок menu, в котором указываем настройки для графика:

добавление точек по событию click, толщину линии (по умолчанию – selected value = 1). В настройке толщины линии применяем элемент <option>  для определения пункта списка контейнера <select>.

1. Создаём контейнер для настройки списка графиков, затем для каждого графика в отдельных блоках прописываем тип (флажок), id, имя и что должно происходить при нажатии на флажок (скрытие графика).
2. Затем при помощи тега canvas создаем области для рисовки графиков в js.
3. Подключаем библиотеку для работы с графиками - <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>\.

Js:

1. Прописываем блок для связки DOM и данных: var app = new Vue({ el: "#app}).
2. Прописываем методы добавления точек и скрытия графиков: в методе добавления точек с помощью свойств target и checked проверяем, отметил ли пользователь флажок (статус checkbox), если значение true, то включаем стиль (pointStyle = true) и обновляем график командой update. В методе скрытия графиков через конструкцию switch-case обращаемся к нужному графику по его имени (прописываем путь plots.children[…], plots получаем по id (const plots = document.getElementById("plots");) и либо удаляем (remove) класс hidden у элемента, либо добавляем его (add)).
3. Добавляем объекты со свойствами (название, данные и ширина линии) для каждого графика.
4. По id обращаемся к каждому графику, записываем в константы и передаем эти константы параметрами при создании экземпляра класса chart для каждого графика.
5. Каждому графику прописываем его тип (line), значения столбцов (labels). В настройках каждого графика (options) прописываем значение false для точек и цвет графика.
6. В конце через свойство addEventListener для тега select при его изменении (change) срабатывает функция: прописываем путь для каждого графика до свойства borderWidth и присваиваем ему полученное значение (e.target.value), обновляем экземпляр класса.

CSS:

1. В универсальном селекторе в свойстве font-family прописываем шрифт для текста на странице, в свойстве box-sizing указываем браузеру учитывать любые границы и внутренние отступы в значениях, которые мы указываем в ширине и высоте элемента, также указываем размер шрифта.
2. В селекторах input и select в свойстве cursor прописываем изменение формы курсора при его наведении на эти селекторы.
3. Для селектора hidden в свойстве display прописываем значение none, чтобы графики не отображались при не нажатой галочке.
4. В селекторе option прописываем настройки цвета, размера и расположения вкладок
5. В селекторе option:hover прописываем настройки для окрашивания кнопки вкладки в синий цвет при наведении на нее курсора
6. В селекторе setting\_down задаём направление главной оси в контейнере и свойство display со значением flex.
7. Аналогично стилизуем остальные элементы страницы