**To-do List Svelte**

*Содржина:*

1. Вовед
2. Преглед на проектот
3. Започнување
4. Структура на проектот
5. Компоненти
6. Државен менаџмент
7. Интерактивност
8. Стајлинг
9. Градење и распоредување
10. Заклучок

**1. Вовед**:

Добредојдовте во документацијата на проектот Svelte Tasker. Оваа документација обезбедува сеопфатен водич за разбирање и работа со проектот Svelte Tasker. Svelte Tasker е веб-апликација која им овозможува на корисниците ефикасно да управуваат со задачите и списоците со задачи. Го користам Svelte framework-от за градење динамичен и одговорен кориснички интерфејс.

На почеток, уште пред да започнам со изработка на проектот, пронајдов неколку туторијали за svelte и svelteKit на YouTube за да ми станат појасни семантиката, логиката, функционалностите и можностите кои ги нуди Svelte framework-от. Потоа започнав со читање на официјалната документацијата на svelte и svelteKit достапна на <https://svelte.dev/docs/introduction> и <https://kit.svelte.dev/docs/introduction> соодветно. Потоа според проектот што го одбраб ми требаше да барам изглед на целата страна како и распределба на елементите на страната. За ова ја искористив страната figma каде што од голем избор на дизајни на страни, го одбрав тој што е најсоодветен за мојата страна и најлесен на окото на корисникот на страната. Откако се одлучив за изгледот на страната, со помош на paint, почнав да ја разделувам страната и да ги определувам компонентите од кои што сакам да ми биде составена страницата.

**2. Преглед на проектот:**

Svelte to-do list е дизајниран да го рационализира управувањето со задачите и да обезбеди интуитивно корисничко искуство. Главните карактеристики на проектот вклучуваат:

* Управување со задачи: Корисниците можат да креираат, уредуваат и бришат задачи.
* Списоци со задачи: Задачите може да се организираат во списоци или категории.
* Респонсивен дизајн: Корисничкиот интерфејс реагира и се прилагодува на различни големини на екранот.
* Интерактивен интерфејс: корисниците можат беспрекорно да комуницираат/манипулираат со задачите и списоците.
* Визуелна повратна информација: апликацијата обезбедува визуелни знаци за статусот на задачите и организацијата на списокот.

**3. Започнување:**

За да започнете со проектот Svelte Tasker, следете ги овие чекори:

1. Клонирајте го складиштето: Клонирајте го складиштето на проектот од изворната локација.
2. Инсталирајте зависности: користете менаџер на пакети (на пр., npm или предиво) за да инсталирате зависности на проектот.
3. Стартувајте го серверот за развој: стартувајте го серверот за развој за да ја прегледате апликацијата локално.
4. Истражувајте и изменете: истражете го изворниот код на проектот и направете ги потребните измени или подобрувања.

**4. Структура на проектот:**

Проектот следи структуриран распоред на директориуми. Клучните директориуми и датотеки вклучуваат:

* src: Го содржи изворниот код на апликацијата.
  + components: Се сместени Svelte компоненти за еднократна употреба.
  + routes: Ги дефинира маршрутите и погледите на апликацијата.
  + stores: Управува со глобалната состојба користејќи ги продавниците Svelte.
* REDME.md file: Што содржи краток опис на апликацијата.
* package.json: Листа на зависности и скрипти на проектот.
* App.svelte: Svelte датотека која ја опишува структурата на веб-апликацијата, вклучувајќи го увозот на различни компоненти, главниот состав на распоредот и место за потенцијални стилови. Таа служи како основа за градење динамична и интерактивна веб-апликација користејќи ја рамката Svelte.
* svelte.config.js: оваа датотека поставува Svelte препроцесирање за овој проект. Користи 'svelte-preprocess' за да конфигурира како Svelte компонентите се обработуваат пред да бидат компајлирани. Стандардната конфигурација е обезбедена од sveltePreprocess(), а дополнителна конфигурација со овозможено PostCSS е зачувана во конфигурацискиот објект. Ова поставување ви овозможува да примените различни чекори за претпроцесирање на вашите Svelte компоненти, како што се ракување со CSS, TypeScript или други препроцесори, за да ги подготвите за компилација.
* Vite.config.ts: Vite конфигурациска датотека напишана во TypeScript. Vite е алатка за градење и сервер за развој што најчесто се користи за градење модерни веб-апликации.
* Index.html: HTML документ кој служи како влезна точка за веб-апликација to-do list изградена со помош на Svelte. Ја поставува основната структура на страницата HTML и вклучува скрипта за вчитување на апликацијата Svelte.
* postcss.config.cjs: конфигурациска датотека за PostCSS, алатка која се користи за обработка на CSS во развој на веб. Поточно, го конфигурира PostCSS да користи два приклучоци: tailwindcss и autoprefixer
* package-lock.json: се користи за одржување на конзистентно и детерминистичко дрво на зависност за проект. Тој игра клучна улога во обезбедувањето дека проектот Svelte, како и секој друг проект Node.js, има постојани зависности во различни средини и кога повеќе програмери работат на проектот.
* tsconfig.node.json: конфигурациска датотека TypeScript која е приспособена за развој специфичен за Node.js. Во суштина, тоа е варијанта на стандардната датотека tsconfig.json, но со поставки оптимизирани за пишување на TypeScript код што ќе работи во околината Node.js.
* app.css: комбинација од CSS и коментари поврзани со tailwindcss и прилагоден стајлинг за криење на лизгачки ленти во веб-апликациите. Ајде да го растуриме

**5. Компоненти:**

Svelte Tasker користи Svelte компоненти за градење на корисничкиот интерфејс. Клучните компоненти вклучуваат:

* **5.1** **MainBody:**
* Овој код претставува управник со прозорец со задачи за веб-апликација. Неговата примарна цел е да им овозможи на корисниците динамично да креираат и управуваат со повеќе прозорци со задачи. Кодот увезува компонента TaskWindow Svelte и обезбедува функционалност за додавање нови прозорци со задачи. Ја увезува компонентата TaskWindow од датотеката „TaskWindow.svelte“. Оваа компонента е одговорна за прикажување на поединечни прозорци со задачи.
* Променливата max се декларира за целите на распоредот и му е доделена висината на екранот на корисникот. Потоа одзема 64 пиксели од моменталната вредност на макс.
* Windows е низа што го содржи почетниот индекс. Се користи за следење на креираните прозорци со задачи.
* индексот е иницијализиран на 1.
* createNewTaskWindow е функција која ја зголемува индексната променлива за 1 и го додава новиот индекс на почетокот на низата на Windows со помош на операторот spread (...). Ова ефективно создава нов прозорец со задачи.
* Ажурираната низа на Windows е најавена на конзолата за цели на отстранување грешки.
* **5.2 ProjectContainer:**
* Кодот им овозможува на корисниците да ја менуваат видливоста на секциите „Проекти“ и „Задачи“. Со кликнување на копчињата до насловите на секциите се проширува или собира соодветната содржина. Оваа интерактивност се постигнува со помош на променливите TypeScript и реактивните карактеристики на Svelte.
* contentProjects и contentTasks ја претставуваат содржината на копчињата за префрлување. Тие првично се поставени на „+“ за да покажат дека содржината е склопена.
* opValueProjects и opValueTasks ја контролираат непроѕирноста на деловите за содржина. Тие првично се поставени на „0“ за целосно транспарентни (скриени).
* heightValueProjects и heightValueTasks: Контролирајте ја висината на деловите за содржина. Тие првично се поставени на „0“ за целосно да се склопи содржината.
* onClickHideTasks и onClickHideProject се Svelte функции активирани со кликнување на копчињата.

Тие ја менуваат содржината на копчињата („+“ или „-“), непроѕирноста (помеѓу 0 и 100) и висината (помеѓу 0 и „автоматско“) на соодветните делови за содржина.

* **5.3 ProjectsPane:**
* Кодот претставува Svelte компонента одговорна за рендерирање на странична лента која прикажува листа на проекти. Страничната лента вклучува заглавие со наслов „Проекти“ и копче за додавање нови проекти.
* Ја увезува компонентата ProjectContainer од датотеката „ProjectContainer.svelte“. Се користи за прикажување на поединечни проектни ставки во страничната лента.
* **5.4 SideBar:**
  + Кодот претставува Svelte компонента одговорна за прикажување на менито за навигација од страничната лента. Страничната лента вклучува ставки за навигација со икони и текст.
  + Оваа компонента ја увезува компонентата SidebarIcon од датотеката „SidebarIcon.svelte“. Најверојатно се користи за прикажување на поединечни икони на страничната лента со текстуални етикети.
* **5.5 SidebarIcon:**
  + Кодот претставува Svelte компонента дизајнирана за прикажување на поединечни икони во менито за навигација на страничната лента. Оваа компонента може да се прилагоди со различни слики и дополнителни својства.
  + Овој фрагмент од код извезува два „props“:
  + imageSource: Го претставува изворот на сликата (URL) за иконата.
  + addProps: Претставува дополнителни својства или класи што може да се додадат во контејнерот со икони.
  + Функцијата onClickHandler е одговорна за справување со настанот за кликнување кога ќе се кликне на иконата. Вклучува порака до конзолата за да покаже дека иконата на страничната лента е кликната.
* **5.6 TaskItem:**
  + Кодот претставува Svelte компонента дизајнирана за прикажување на картички со ставки. Секоја картичка прикажува информации за одредена ставка, вклучувајќи го нејзиниот наслов, поднаслов и лента за напредок.
* **5.7 TaskWindow:**
  + Кодот претставува Svelte компонента дизајнирана за прикажување листа на задачи. Корисниците можат да креираат нови задачи со внесување имиња на задачи и кликнување на копчето „Нова задача“. Списокот со задачи прикажува поединечни задачи користејќи ја компонентата TaskItem.
  + Оваа компонента ја внесува компонентата TaskItem од датотеката „TaskItem.svelte“. Најверојатно се користи за прикажување на поединечни ставки за задачи во списокот со задачи.
  + index: променлива што се користи за следење на индексот на задачи. Иницијализиран е на 1.
  + items: низа што се користи за складирање на ставки за задачи. Првично, содржи една ставка со индекс 1.
  + createNewTask(): Функција која додава нова задача во низата со ставки. Го зголемува индексот и го поставува новиот индекс на низата со ставки, ефективно создавајќи нова задача.
* **5.8 TopBar:**
  + Кодот претставува Svelte компонента дизајнирана за прикажување на горната лента на врвот на веб-страницата. Горната лента прикажува кориснички информации, икони и некои интерактивни елементи.
  + Оваа компонента ја увезува компонентата TopBarIcon од датотеката „TopBarIcon.svelte“. Се користи за прикажување на поединечни икони во горната лента.
  + name: променлива која го содржи името на корисникот. Тој е иницијализиран со вредноста „Daniel“.
  + now: Променлива која го зачувува тековниот датум како форматирана низа користејќи го методот toLocaleDateString.
* **5.9 TopBarIcon:**
  + Кодот претставува Svelte компонента дизајнирана за прикажување на поединечни икони во горната лента или менито за навигација. Оваа компонента може да се прилагоди со различни извори на слика и дополнителни атрибути.
  + Овој фрагмент од код извезува два „реквизити“:
    - imgSource: Го претставува изворот на сликата (URL) за иконата.
    - extraAtr: Претставува дополнителни атрибути или класи што може да се додадат во контејнерот со икони.
  + Функцијата onClickHandler е одговорна за справување со настанот за кликнување кога ќе се кликне на иконата.

Запишува порака до конзолата за да покаже дека иконата е кликната.

* **5.10 App.svelte:**
  + Овие изјави за увоз внесуваат Svelte компоненти од нивните соодветни датотеки. Секоја компонента претставува специфичен дел од корисничкиот интерфејс на веб апликацијата:
  + MainBody: Ја претставува главната област за содржина на апликацијата.
  + ProjectsPane: Претставува дел за прикажување на содржини поврзани со проектот.
  + Странична лента: Претставува странична лента или мени за навигација.
  + TopBar: Претставува врвна лента за навигација или заглавие.

**6. State Management:**

* **6.1 taskStore:**
  + Содржи еден ред код:

import { writable } from "svelte/store"; -> кој што се користи во Svelte апликации за внесување на функцијата за запишување од модулот „svelte/store“. Тоа е клучен дел од управувањето со состојбата и реактивноста во компонентите Svelte.

* + Функцијата writeable е дел од системот за реактивни складишта на Svelte, кој ви овозможува да креирате и управувате со реактивни податоци што може да се споделат низ компонентите. Поточно, writable се користи за создавање writeable store. Еве преглед на неговата намена:
    - „writeable store“ е тип на продавница во Свелте што има променлива вредност. Може да се чита од и да се запишува со една или повеќе компоненти. Кога вредноста во writeable store, ќе се промени, сите компоненти што се претплатени на неа автоматски повторно ќе се прикажуваат за да ја одразат новата вредност.
* **6.2 viewStore:**
  + Го содржи истиот код и како taksStore

**7. Стајлинг:**

Проектот го користи tailwind css framework-от за стајлинг. Следи пристап кон дизајн кој реагира за да обезбеди конзистентно и пријатно искуство за корисниците на различни уреди. Стилот е модуларен и организиран за лесно одржување и прилагодување.

Датотеката tailwind.config.cjs претставува конфигурациска датотека за Tailwind CSS framework-от. Накратко, овој фрагмент од код го конфигурира Tailwind CSS за проект. Тој одредува кои датотеки треба да се обработуваат за да се генерираат стилови (датотеки HTML, JavaScript, Svelte и TypeScript во директориумот './src/'), но не ја проширува или прилагодува стандардната тема и не вклучува никакви прилагодени приклучоци. Оваа конфигурација е основно поставување за користење на Tailwind CSS во проект и може дополнително да се приспособи по потреба.

Во Директориумот assets се наоѓаат сите слики кои ке се користат во самиот проект.

**8. Заклучок:**

Проектот Svelte to-do list ги прикажува можностите на frameworkot Svelte со TypeScript за градење динамични веб-апликации. Тој обезбедува кориснички интерфејс за управување со задачи и служи како практичен пример за веб-развој со користење на овие технологии. Може да се направат дополнителни подобрувања и прилагодувања за да се прилагоди проектот на специфични барања и преференции.