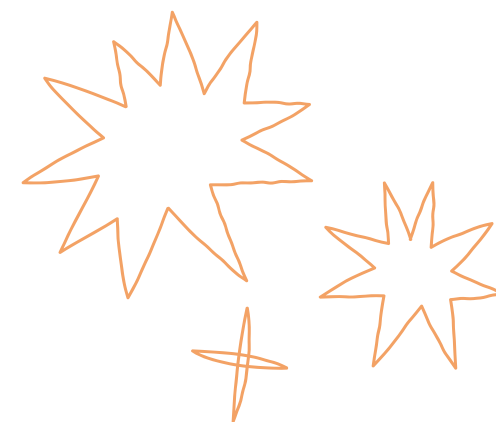
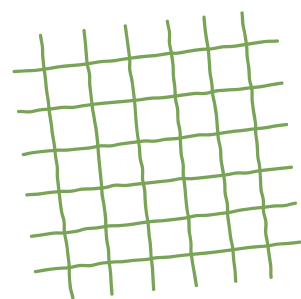
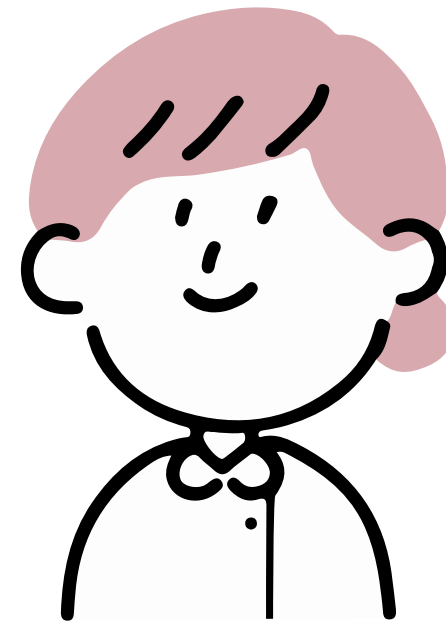


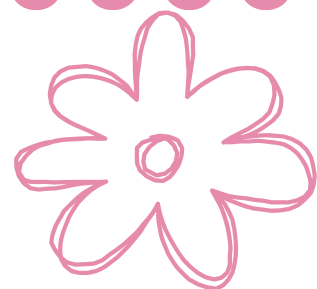


おけまるの 投資開発室

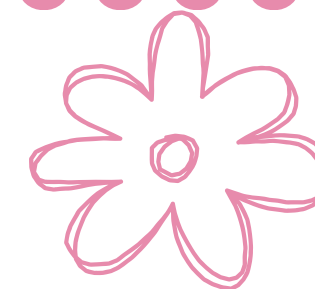


おけまるです





前回までの

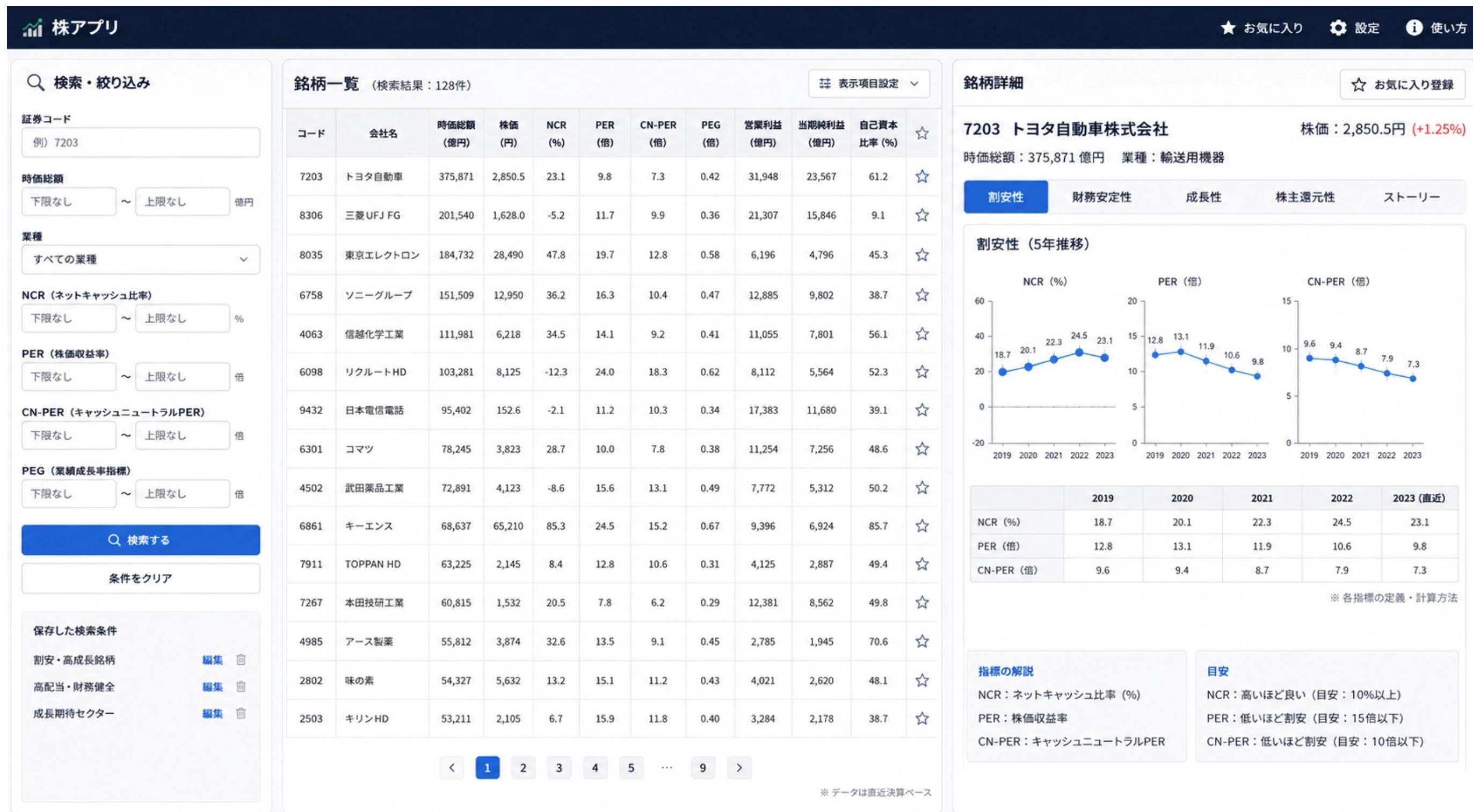


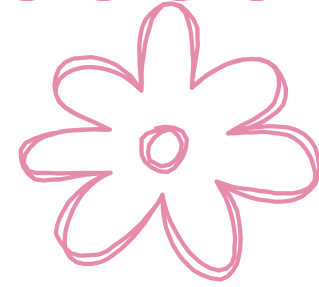
アプリ作成のきっかけ

- ...
- もう働きたくない
- 株をやろう
- 株のアプリをつかって、それをもとに株をやろう

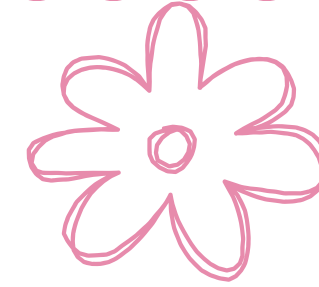


アプリのイメージ図

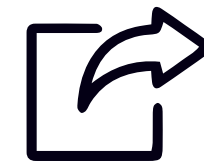




設計



- MVP（必要な）機能の決定
 - 銘柄一覧を見る
 - 銘柄を条件で絞り込む
 - 銘柄詳細を見る
 - 割安、財務安定性、成長性、ストーリーなど
- ER図の作成
 - データの持ち方の整理
- フロー図の作成
 - データの流れ



今回実装したもの

- Next.jsで3カラム画面
- タブ切り替えを作成



完成イメージ：銘柄詳細タブ

銘柄詳細 ☆ お気に入り登録

7203 トヨタ自動車株式会社 株価：2,850.5円 (+1.25%)

時価総額：375,871 億円 業種：輸送用機器

割安性 財務安定性 成長性 株主還元性 ストーリー

割安性（5年推移）

NCR（%）	PER（倍）	CN-PER（倍）
--------	--------	-----------

3カラムレイアウト



```
.mainLayout {  
  display: grid;  
  grid-template-columns: 280px 1fr 520px;  
  gap: 12px;  
}
```

コンポーネント構成

components/

Header/

SearchPanel/

StockList/

StockDetail/

tabs/

ValueTab.tsx

StabilityTab.tsx

GrowthTab.tsx

ShareholderReturnTab.tsx

StoryTab.tsx

銘柄検索

銘柄一覧

銘柄詳細

割安性

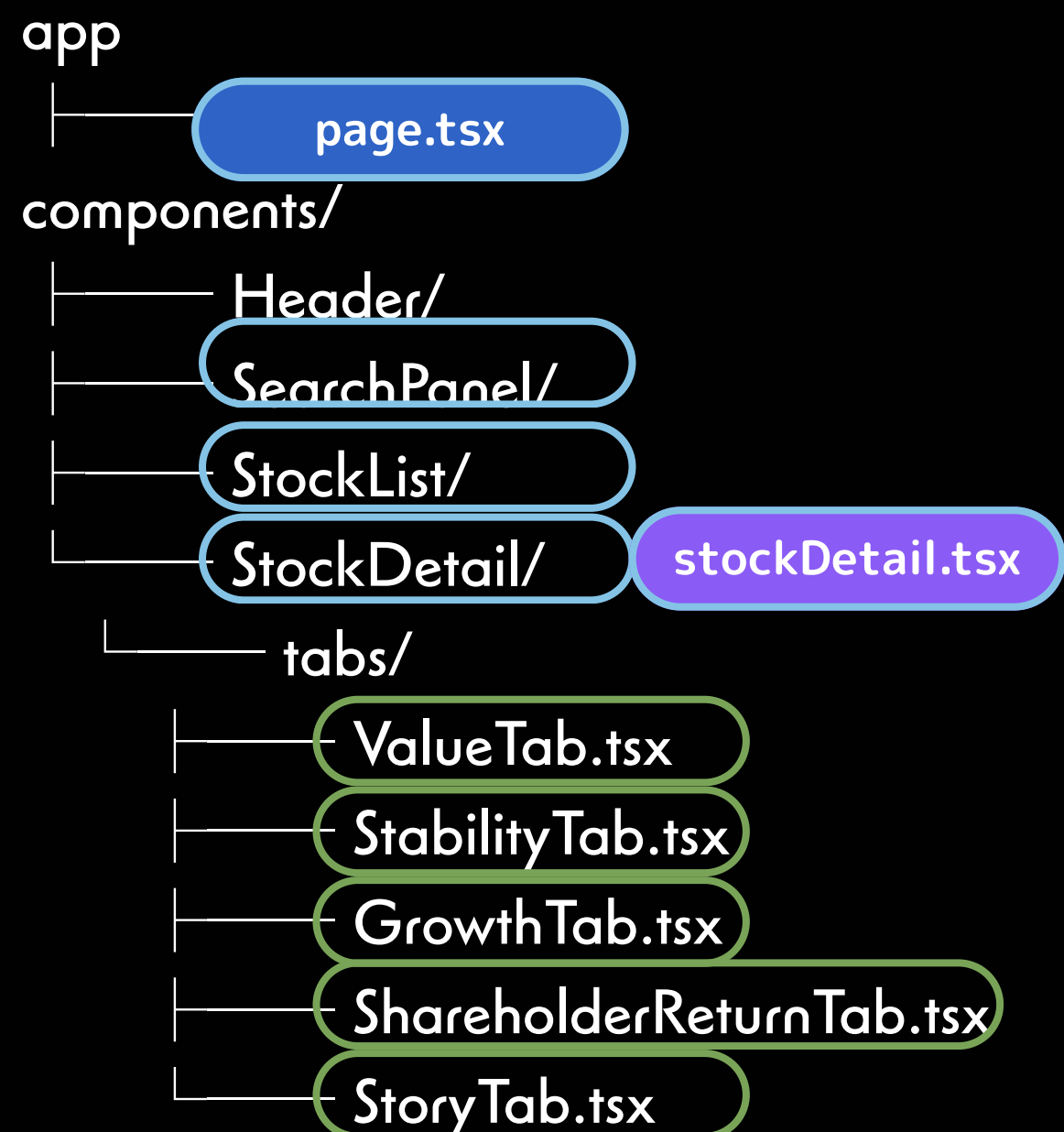
成長性

財務安定性

株主還元性

ストーリー

コンポーネント構成



役割の分け方

page.tsx

各部品の配置を担当

StockDetail.tsx

タブの状態管理を担当

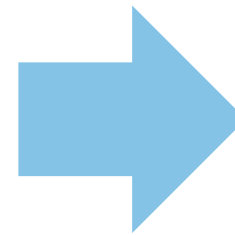
各Tab

タブごとの内容を担当

配列からタブを作成

配列を作成

```
const tabs = [  
  { id: "value", label: "割安性" },  
  { id: "stability", label: "財務安定性" },  
  { id: "growth", label: "成長性" },  
  { id: "shareholderReturn", label: "株主還元性" },  
  { id: "story", label: "ストーリー" },  
];
```



配列のデータをボタンに変換

```
{tabs.map((tab) => (  
  <button key={tab.id}>  
    {tab.label}  
  </button>  
))}
```

クリックでタブを切り替える仕組み



`const [状態変数, 状態を更新する関数] = useState(初期値) ;`

`onClick={() => setActiveTab(tab.id)}`

ボタンが押されたら、`setActiveTab`関数を実行する

```
const [activeTab, setActiveTab]
= useState<TabName>("value");
```

```
onClick={() => setActiveTab(tab.id)}
```

```
{activeTab === "growth" && <GrowthTab />}
```

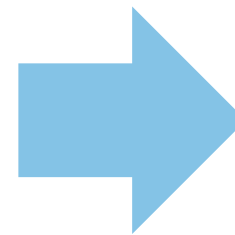
`const [状態変数, 状態を更新する関数] = useState(初期値) ;`

選択中のタブにCSSを適用

タブが押されていたら、
classNameをactiveTabにする

- tabsの中
- button要素のactiveTab
であれば適用

```
<button  
  key={tab.id}  
  type="button"  
  onClick={() => setActiveTab(tab.id)}  
  className={activeTab === tab.id ?  
styles.activeTab : ""}  
>
```



```
className={  
  activeTab === tab.id  
  ? styles.activeTab  
  : ""  
}
```

```
.tabs button.activeTab {  
  color: #ffffff;  
  background-color: #2563d9;  
}
```

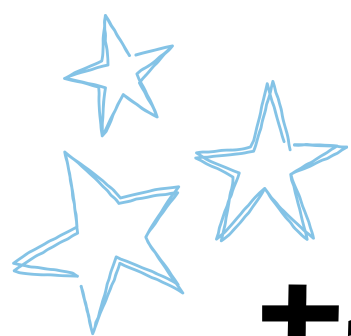
今回学んだこと・次回

今回学んだこと

- コンポーネントの役割分担
- CSSGridによる3カラムレイアウト
- 配列のmapを使ったボタン作成
- useStateを使ったアクティブタブの状態管理
- クリックイベントによる内容とレイアウト（色）の切り替え

次回

- 検索条件のレイアウト作成
- 銘柄一覧表示のレイアウト作成



おけまるチャンネルを
よろしくおねがいします！

