Ionic generate page AddNewItem

Ionic generate page UpdateItem

Za svaki zadatak bi trebalo da se prikaže:

• Naziv zadatka

• Kategorija zadatka (da li je niskog, srednjeg ili visokog prioriteta)

• Datum za izvršavanje zadatka

1. front kostur: home.page.html

<ion-header>

  <ion-toolbar>

    <ion-title>

      <h1>Today</h1>

      <p class="date">Datum...</p>

    </ion-title>

  </ion-toolbar>

</ion-header>

<ion-content [fullscreen]="true">

  <!-- prodji kroz listu svih taskova -->

  <ion-list>

    <ion-card>

      <!-- header kartice -->

      <ion-card-header>

        <ion-card-title>Naziv aktivnosti..</ion-card-title>

        <ion-item>

          <ion-icon>Ikonica..</ion-icon>

          <ion-label>Kategorija aktivnosti...</ion-label>

        </ion-item>

      </ion-card-header>

      <!-- sadrzaj kartice -->

      <ion-card-content>

        <p>Datum aktivnosti...</p>

        <ion-row>

          <ion-button>Update</ion-button>

          <ion-button>Delete</ion-button>

        </ion-row>

      </ion-card-content>

    </ion-card>

  </ion-list>

</ion-content>

<https://ionicframework.com/docs/api/header#translucent-header>

<https://ionicframework.com/docs/api/content>

2. Ajmo logika> home.page.ts:  
Potrebno je kreirati logiku za ovu stranicu. Logika aplikacije se uvek piše u .ts fajlovima odgovarajuće stranice. U fajlu home.page.ts ćemo pre svega izvršiti inicijalizaciju potrebnih promenljivih:

PRVO DEFINISEMO TASK I INIJALIZUJEMO TRENUTNI DATUM I NIZ TASKOVA>

import { Component } from '@angular/core';

type Task = {

  name?: string,

  date?: Date,

  category?: string

}

@Component({

  selector: 'app-home',

  templateUrl: 'home.page.html',

  styleUrls: ['home.page.scss'],

})

export class HomePage {

  today: number = Date.now();

  tasks: Array<Task> = [];

  constructor() {}

}

Onda odemo na front:  
**Date pipe** se primenjuje kako bi se formatirala vrednost promenljive today kao pun datum koristeći format specifikator 'fullDate'. Dobijeni formatirani datum se zatim prikazuje unutar paragrafa.

<p class="date">{{today | date:'fullDate'}}</p>

Tako komunicira front i bek.

Bek:

 // izvrsava se svaki put prilikom inicijalizacije komponente

  ngOnInit(): void {

    this.tasks = [

      {

        name: "Uraditi MPOS domaci",

        date: new Date(),

        category: 'Low'

      },

      {

        name: "Uraditi EEP domaci",

        date: new Date(),

        category: 'High'

      },

    ];

  }

FRONT:

<https://ionicframework.com/docs/api/icon>

 <!-- prodji kroz listu svih taskova -->

  <ion-list \*ngFor="let task of tasks">

    <ion-card>

      <!-- header kartice -->

      <ion-card-header>

        <ion-card-title>{{task.name}}</ion-card-title>

        <ion-item>

          <ion-icon \*ngIf="task.category== 'Low'" name="logo-ionic" color="warning"></ion-icon>

          <ion-icon \*ngIf="task.category== 'High'" name="logo-ionic" color="danger"></ion-icon>

          <ion-label>&nbsp;{{task.category}}</ion-label>

        </ion-item>

      </ion-card-header>

      <!-- sadrzaj kartice -->

      <ion-card-content>

        <p>{{task.date | date: 'fullDate'}}</p>

        <ion-row>

          <ion-button>Update</ion-button>

          <ion-button>Delete</ion-button>

        </ion-row>

      </ion-card-content>

NA KRAJU HOCEMO DA OMOGUCIMO DA SE SA HOME STRANICE OTVARAJU MODALI ZA KREIRANJE NOVE STRANICE I ZA UPDATE TASKA

NA FRONTU, DODAMO DUGME ZA KREIRANJE NOVOG TASKA>

Na kraju, na početnoj stranici ćemo dodati dugme pomoću FAB (Floating Action Button) komponente koje služi za dodavanje novog elementa na interfejsu, u našem slučaju novog zadatka.

  <ion-fab vertical="bottom" horizontal="end" slot="fixed">

    <ion-fab-button>

      <ion-icon name="add">Add</ion-icon>

    </ion-fab-button>

  </ion-fab>

NA BEKU KREIRAMO MODALE> PRVO U KONCTURKTORU  
constructor(public modalCtrl: ModalController) - Konstruktor klase koji prima "modalCtrl" objekat kako bi bio dostupan u komponenti. Ovo omogućava korišćenje "ModalController" za otvaranje modala na ovoj stranici

  constructor(public modalCtrl: ModalController) {}

i kreiramo funkcije:

  async goToAddPage() {

    const modal = await this.modalCtrl.create({

      component: AddNewItemPage

    });

    return await modal.present();

  }

Pa na frontu pozovemo preoko dogadjaja click:

   <ion-fab-button (click)="goToAddPage()">

Update:

  async goToUpdatePage(task: Task) {

    const modal = await this.modalCtrl.create({

      component: UpdateItemPage,

      componentProps: task

    });

    return await modal.present();

  }