**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО»**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

по дисциплине «Современные технологии *Frontend* разработки»

на тему: «*Front-End* фреймворки»

Выполнил: студент гр. ИТП-41

Леоненко В.Ю.

Принял: преподаватель

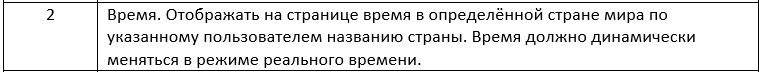
Ястребов А.А.

Гомель 2023

**Цель работы:** изучить *Front-End* фреймворки.

**Задание**

Разработать интерфейс веб-приложения с использованием *frontend*-фреймворков *React*, *Angular* или *Vue* (на выбор).



В зависимости от задачи, можно как создавать собственные словари / шаблоны, так и пользоваться общедоступными бесплатными *API / json* различных сервисов.

Разработанное приложение должно иметь:

• Шапку с логотипом (произвольным, можно текстовым) и темой работы;

• Футер (с подписью студента, который разрабатывал веб-приложение);

• Основная область веб-приложения.

**Ход работы**

В ходе данной лабораторной работы разработано *Angular* приложение, которое обращается к *World Time API* с помощью *HTTP* запросов для получения списка поддерживаемых временных зон и времени для выбранной пользователем временной зоны.

На рисунке 1 представлен графический интерфейс приложения.

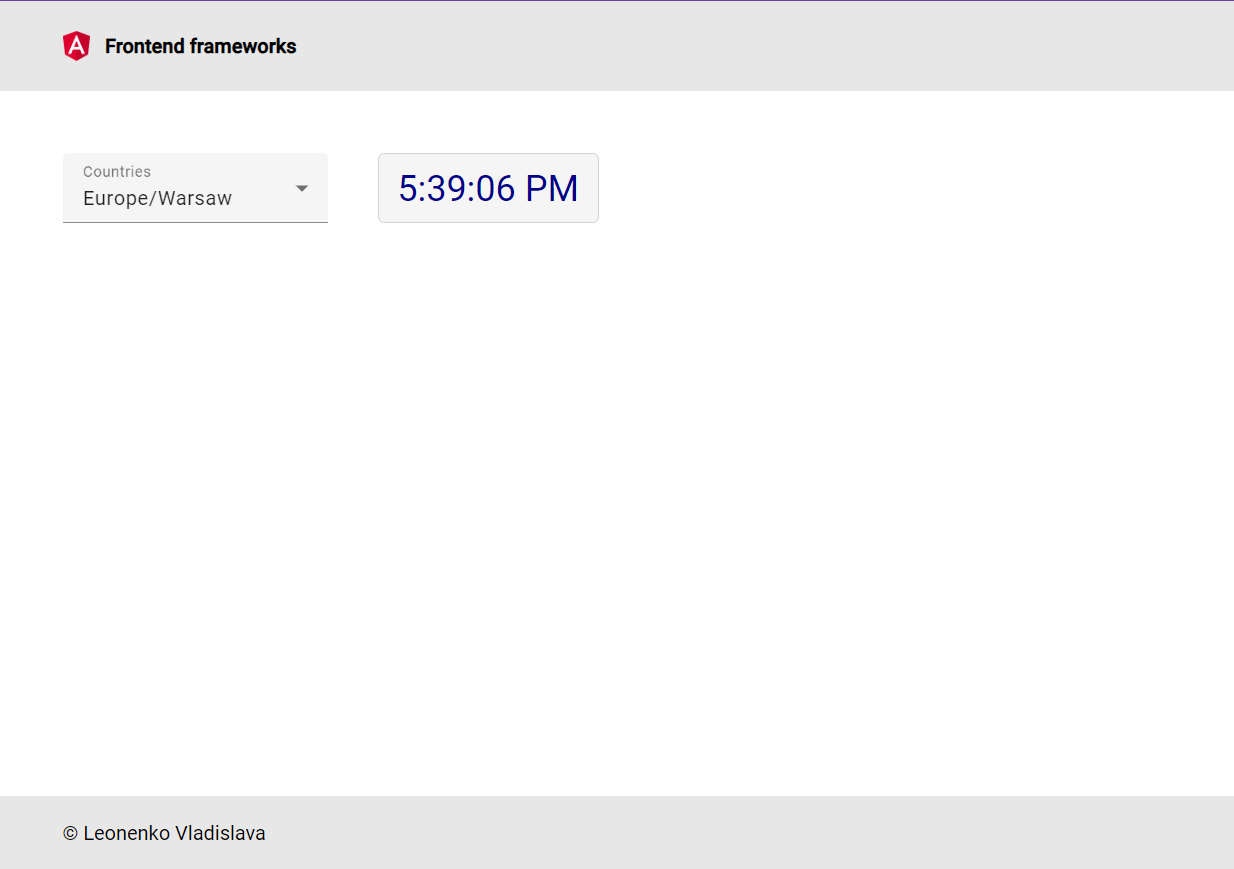


Рисунок 1 – Графический интерфейс приложения

Текст программы представлен в приложении А.

**Вывод:** в результате выполнения лабораторной работы изучен *Front-End* фреймворк *Angular*.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

WorldTimeService.ts

import { HttpClient } from '@angular/common/http';

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

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class WorldTimeService {

  private apiUrl = 'http://worldtimeapi.org/api/timezone';

  constructor(private http: HttpClient) {}

  getTimeForCountry(country: string): Observable<any> {

    const url = `${this.apiUrl}/${country}`;

    return this.http.get(url);

  }

  getCountries(): Observable<string[]> {

    const url = `${this.apiUrl}/`;

    return this.http.get<string[]>(url);

  }

}

HomeComponent.ts

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

import { WorldTimeService } from 'src/app/services/world-time.service';

@Component({

  selector: 'app-home',

  templateUrl: './home.component.html',

  styleUrls: ['./home.component.scss']

})

export class HomeComponent implements OnInit {

  countries = ["Europe/Warsaw"]

  currentTime: Date | undefined

  currentCountry = "Europe/Warsaw"

  constructor(private worldTimeService: WorldTimeService) {}

  ngOnInit() {

    this.worldTimeService.getCountries().subscribe(data => this.countries = data)

    this.worldTimeService.getTimeForCountry(this.currentCountry).subscribe((data) => {

      this.currentTime = new Date(data.datetime.slice(0, 19));;

      this.updateTime()

    });

  }

  onSelectionCahnge() {

    this.worldTimeService.getTimeForCountry(this.currentCountry).subscribe((data) => {

      this.currentTime = new Date(data.datetime.slice(0, 19));

    });

  }

  updateTime() {

    this.currentTime = new Date(Number(this.currentTime?.getTime()) + 1000)

    setTimeout(() => this.updateTime(), 1000);

  }

}