

GOVERNMENT POLYTECHNIC, AHMEDABAD
COMPUTER ENGINEERING DEPARTMENT



**Affiliated
To
Gujarat Technological University, Ahmedabad**

Micro project Report
D. E. second Year (Semester–IV)
Sub: Modern Practical Tools (4340705)



Government Polytechnic, Ahmedabad
Computer Engineering Department

CERTIFICATE

This is to certify that

| Sr. No. | Enrollment No. | Name |
|---------|----------------|-------------------------|
| 1 | 226170307235 | Yadav Prajwal Premsagar |
| 2 | 226170307228 | Vatsal Vaghela |
| 3 | 226170307233 | Vohra Moin |
| 4 | 2261703072225 | Vandra Sanket |
| 5 | 226170307206 | Suthar Ridham B. |

Of **Fourth** semester of Diploma in Computer Engineering of Government Polytechnic, Ahmedabad has completed the Micro-Project satisfactorily in Subject **Modern Practical Tools (4340705)** for the academic year **2024 -2025** as prescribed in the curriculum.

Lecturer,
Computer Engg. Dept.,
Government Polytechnic, Ahmedabad

HOD
Computer Engg. Dept.,
Government Polytechnic,
Ahmedabad

RUBRICSFORMICRO-PROJECTASSESSMENT

| Parameters | Allocated Marks | High | Medium | Low |
|---|-----------------|--|--|--|
| Problem Completeness/ Correctness (R1) | 8 | Problem is Properly Analyzed and explained | Problem is Properly Analyzed but Partially explained | Problem is Properly Analyzed but not Solved. |
| | | 8Marks | 5Marks | 2Marks |
| Viva Voce(R2) | 2 | Student Answered All The Viva Voce Questions | Student Answered Only A Few Viva Voce Questions | Student Did Not Answer Any Viva Voce Questions |
| | | 2Marks | 1Marks | 0Marks |

INDEX

1. introduction of project
2. Solution of Problem/ Explanation/ Conclusion
3. References if any

| Enrollment Number | Student Name | Marks(R1) | Marks(R2) | Total Marks |
|---------------------------|-------------------------|-----------|-----------|-------------|
| 226170307235 | Yadav Prajwal Premsagar | | | |
| 226170307228 | Vatsal Vaghela | | | |
| 226170307233 | Vohra Moin | | | |
| 2261703072225 | Vandra Sanket | | | |
| 226170307206 | Suthar Ridham B. | | | |
| Name and Sign of Faculty: | | | | |

Project Discription

Currency Converter Application

Overview

The Currency Converter Application is a web-based tool built using the Angular framework. It allows users to convert amounts between different currencies using real-time exchange rates fetched from a public API. This application is designed to be user-friendly and provides quick and accurate currency conversions.

Features

- **Real-Time Exchange Rates:** Fetches the latest exchange rates from a reliable API.
- **Multiple Currencies:** Supports a wide range of currencies for conversion.
- **User Input:** Allows users to input the amount they wish to convert.
- **Simple Interface:** Clean and intuitive user interface for easy navigation.
- **Conversion Result:** Displays the converted amount immediately upon request.

Technologies Used

- Angular: A popular web application framework used for building the front-end of the application.
- HttpClientModule: Used for making HTTP requests to fetch exchange rates.
- ExchangeRate-API: A third-party API service providing real-time exchange rates.

Project Structure

- Components:
 - ConverterComponent: Handles the user interface and conversion logic.
- Services:
 - CurrencyService: Manages fetching exchange rates from the API.

Implementation Details

1. Setup and Initialization:

- Created a new Angular project using Angular CLI.
- Installed necessary packages such as HttpClientModule.

2. Service to Fetch Exchange Rates:

- Implemented CurrencyService to handle HTTP requests to the ExchangeRate-API.
- API endpoint configured to fetch the latest exchange rates.

3. Converter Component:

- Developed ConverterComponent to provide the UI and handle user interactions.
- Template includes dropdowns for selecting currencies, an input for the amount, and a button to trigger the conversion.
- Conversion logic calculates the converted amount based on the fetched exchange rates.

4. User Interface:

- Simple and clean design with a central container to hold all elements.
- Form elements include currency selectors and an amount input field.
- Result displayed dynamically after conversion.

How to Run the Application

1. Install Angular CLI:

```
bash
npm install -g @angular/cli
```

2. Create and Navigate to the Project:

```
bash
ng new currency-converter
cd currency-converter
```

3. Generate Components and Services:

```
bash
ng generate component converter
ng generate service currency
```

4. Implement Service and Component Logic:

- Add HTTP client logic to currency.service.ts.
- Add component logic to converter.component.ts.
- Update the component template in converter.component.html.
- Apply basic styling in converter.component.css.

5. Run the Application:

```
bash
ng serve
```

6. Access the Application:

- Open a web browser and navigate to <http://localhost:4200/>.

Conclusion

The Currency Converter Application is a practical example of using Angular to build a web-based tool that interacts with a third-party API. It provides a straightforward user experience for converting currencies, demonstrating the power of Angular in creating dynamic and responsive web applications.

GitHub Link of Project

<https://github.com/PrajwalYadav07/Angular-Currency-Converter>

Some Main Codes of the project

app.component.html

```

<div class="container">
  <div class="card" >
    <h1>Currency Converter</h1>
    <div *ngIf="isDataAvailable">
      <form name="form-exchange" (ngSubmit)="onSubmit()">
        <div class="form-currency" #formExchange (window:resize)="windowResize()">
          <label>Amount</label>

          <div>
            <div class="input-amount input-field">
              <span class="prefix">{{from_symbol}}</span>
              <input required lang="en-US" type="number" [(ngModel)]="amount_value"
(change)="changeAmountValue()" name="amount" (focusout)="changeAmountValue()"
#amount_input placeholder="Amount" step="0.01" >
            </div>

          </div>
          <label>From</label>

          <app-currency-selector #from [selectorId]='from'
[changeCurrency]="selectFrom"></app-currency-selector>

        <div></div>

        <button class="switch-btn" type="button" (click)="switchCurrencies()">
          <i class="bi bi-arrow-left-right"></i>
        </button>

        <label>To</label>

        <app-currency-selector #to [selectorId]='to' [changeCurrency]="selectTo"></app-
currency-selector>
      </div>

      <div class="submit-btn" [style.width.px]="formExchange.offsetWidth" #submitBtn
*ngIf="!isResult">
        <button type="submit">Convert</button>
      </div>

    </form>
    <div class="result-currency" *ngIf="isResult">
      <div class="result-from">
        {{resultFrom}}
      </div>

```



```

        <div class="result-to">
            {{resultTo}}
        </div>
        <div class="result-info">
            {{resultInfo}}
        </div>
    </div>
</div>
<div class="lds-ellipsis" *ngIf="!isDataAvailable &&
!failedToLoad"><div></div><div></div><div></div><div></div></div>

<div *ngIf="failedToLoad" [style.text-align]='center'>
<h2>
    <i class="bi bi-bug-fill"></i></h2>
<h3>Failed to load exchange rate from the server</h3>
</div>

<div class="lastUpdate" *ngIf="isResult">
    Last updated: {{lastUpdate}}
</div>

<div class="madeBy">Made by Prajwal Yadav</div>

</div>

</div>

```

CSS Style

```

.btn, input, .input-field, .switch-btn, button{
    width: 260px;
    height: 50px;
    border-radius: 5px;
    text-align: left;
    justify-content: center;
    padding: 8px;
    font-weight: 200;
    letter-spacing: 1px;
    //text-transform: uppercase;
    background-color: white;
    -webkit-box-shadow: 0px 0px 19px -12px rgba(0, 0, 0, 0.5);
    -moz-box-shadow: 0px 0px 19px -12px rgba(0, 0, 0, 0.5);
    box-shadow: 0px 0px 19px -12px rgba(0, 0, 0, 0.5);
    outline: 2px solid #d2d2d2;
    border: none;

    &:focus{
        background-color: white;
        outline: 2px solid rgb(153, 153, 245);
    }

    &:hover{
        background-color: white;
        outline: 2px solid rgb(153, 153, 245);
        border: none;
    }
}

```

```

a, i{
  &.bi{
    color:  rgb(153, 153, 245);
  }
}
}

```

```

a, i{
  &.bi{
    color:  rgba(82, 82, 82, 1);
  }
}
}

```

Main.ts

```

import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));

```

currency-services.component.ts

```

import {Injectable} from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Currency } from 'src/app/Currency';
//import { CURRENCIES } from 'src/app/mock-currency';

@Injectable({
  providedIn: 'root',
})
export class CurrencyServiceComponent{

  private currencies:Currency[] = [];
  private lastUpdate;
  constructor(private http: HttpClient) {
  }

  public getCurrencies(){
    return this.currencies;
  }

  public getLastUpdate(){
    return this.lastUpdate;
  }
}

```

```

public getCurrenciesPromise() {
  return new Promise<any>((resolve, reject) => {
    if(this.currencies.length==0)
    {
      this.http.get<any>('https://open.er-api.com/v6/latest/USD').subscribe(data => {
        for (var key in data.rates){
          var value = data.rates[key];
          let currency:Currency = {rate: value, full_name: '', name: key, symbol: ''};
          this.currencies.push(currency);
        }
        this.lastUpdate = data.time_last_update_utc;
      });
      this.http.get<any>('https://restcountries.com/v3.1/all?fields=currencies').subscribe(data => {
        data.forEach(currency => {
          let name = Object.keys(currency.currencies)[0]
          var index = this.currencies.findIndex(element => element.name==name);
          if (index!=-1)
            this.currencies[index] = {...this.currencies[index], full_name:
currency.currencies[name].name, symbol: currency.currencies[name].symbol}
          }
        )
        resolve(this.currencies);
      },
      () => {
        reject();
      }
    ),
    () => {
      reject();
    }
  )
  }
  else {
    resolve(this.currencies);
  }
  })}
}

```

Test.ts

// This file is required by karma.conf.js and loads recursively all the .spec and framework files

```

import 'zone.js/testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

```

```
declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    <T>(id: string): T;
    keys(): string[];
  };
};
```

```
// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
);
```

```
// Then we find all the tests.
const context = require.context('./', true, /\.spec\.ts$/);
// And load the modules.
context.keys().forEach(context);
```

angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "currency-exchange": {
      "projectType": "application",
      "schematics": {
        "@schematics/angular:component": {
          "style": "scss"
        }
      },
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "docs",
            "index": "src/index.html",
            "main": "src/main.ts",
            "polyfills": "src/polyfills.ts",
            "tsConfig": "tsconfig.app.json",
            "inlineStyleLanguage": "scss",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "node_modules/bootstrap/scss/bootstrap.scss",
              "node_modules/bootstrap-icons/font/bootstrap-icons.css",
              "src/styles.scss"
            ],
            "scripts": [
              "node_modules/@popperjs/core/dist/umd/popper.min.js",
              "node_modules/jquery/dist/jquery.min.js",
```

```

      "./node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
    ]
  },
  "configurations": {
    "production": {
      "budgets": [
        {
          "type": "initial",
          "maximumWarning": "500kb",
          "maximumError": "1mb"
        },
        {
          "type": "anyComponentStyle",
          "maximumWarning": "2mb",
          "maximumError": "5mb"
        }
      ],
      "fileReplacements": [
        {
          "replace": "src/environments/environment.ts",
          "with": "src/environments/environment.prod.ts"
        }
      ],
      "outputHashing": "all"
    },
    "development": {
      "buildOptimizer": false,
      "optimization": false,
      "vendorChunk": true,
      "extractLicenses": false,
      "sourceMap": true,
      "namedChunks": true
    }
  },
  "defaultConfiguration": "production"
},
"serve": {
  "builder": "@angular-devkit/build-angular:dev-server",
  "configurations": {
    "production": {
      "browserTarget": "currency-exchange:build:production"
    },
    "development": {
      "browserTarget": "currency-exchange:build:development"
    }
  },
  "defaultConfiguration": "development"
},
"extract-i18n": {
  "builder": "@angular-devkit/build-angular:extract-i18n",
  "options": {
    "browserTarget": "currency-exchange:build"
  }
},
"test": {
  "builder": "@angular-devkit/build-angular:karma",
  "options": {
    "main": "src/test.ts",

```

```

    "polyfills": "src/polyfills.ts",
    "tsConfig": "tsconfig.spec.json",
    "karmaConfig": "karma.conf.js",
    "inlineStyleLanguage": "scss",
    "assets": [
      "./src/favicon.ico",
      "./src/assets"
    ],
    "styles": [
      "./node_modules/bootstrap/scss/bootstrap.scss",
      "./node_modules/bootstrap-icons/font/bootstrap-icons.css",
      "../node_modules/bootstrap/dist/css/bootstrap.min.css",
      "./src/styles.scss"
    ],
    "scripts": [
      "./node_modules/@popperjs/core/dist/umd/popper.min.js",
      "./node_modules/jquery/dist/jquery.min.js",
      "./node_modules/bootstrap/dist/js/bootstrap.bundle.min.js",
    ]
  }
}
}
},
"cli": {
  "analytics": false
}
}

```

Tsconfig.app.json

```

/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/app",
    "types": []
  },
  "files": [
    "src/main.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}

```

Ontput :-

