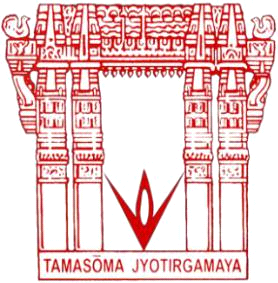
**Department of CSE – Cyber Security**

**Report on Course Based Project**

**CURRENCY CONVERTER**

**Team Members:**

21071A6208 – Hari Boppana

21071A6215 – Amulya Dharavath

**Course Instructor Name : Mrs. E. Lalitha**

Assistant Professor, VNRVJIET

**AIM:**

To create a program that can give us real time currency exchange rates.

**PROBLEM STATEMENT:**

The problem is to develop a user-friendly, reliable, and accessible real-time currency exchange rates application that provides accurate currency rates information to users on multiple platforms.

**SOFTWARE REQUIREMENT SPECIFICATION (SRS):**

|  |  |
| --- | --- |
| **S. No** | **Software Requirement Specification** |
|  |  |
|  |  |
| 1.0 | Introduction |
|  |  |
| 1.1 | Purpose |
|  |  |
| 1.2 | Scope |
|  |  |
| 1.3 | Technologies to be used |
|  |  |
| 1.4 | Tools to be used |
|  |  |
| 1.5 | Overview |
|  |  |
| 1.6 | Product Perspective |
|  |  |
| 1.7 | Interface |
|  |  |
| 1.8 | System Functions |
|  |  |
| 1.9 | User Characteristics |
|  |  |
| 2.0 | Future Scope |
|  |  |
| 2.1 | UML Diagrams( Activity Diagram ) |
|  |  |
| 2.2 | Source Code & Output |
|  |  |

* **INTRODUCTION**

A currency converter is an application that helps with the quick and easy conversion of currencies on the basis of exchange rates. In simple language, a currency converter helps in converting an amount from one currency to another currency. A currency converter is generally used while traveling abroad, by businessmen, and while exporting and importing trades.

**1.1 PURPOSE**

Currency conversion is of practical use to tourists who travel abroad, to businesses who do business overseas or are involved in imports and exports, and to FX traders. A currency converter is an web application that allows for the quick conversion of any currency into any other currency.

One can use a converter to perform entire tasks of calculations. Apart from performing calculations, this tool will always give you regular updates on currency rates. It would be very difficult for investors to convert different currencies without using a currency converter.

* **SCOPE**
* **Currency Exchange Rates Collection**: Gathering currency exchange rates data from various countries through API.
* **User Interface:** Developing a user-friendly interface that allows users to easily convert currency into desired country’s currency.
* **User Support:** Providing support for users, including troubleshooting, answering questions, and addressing concerns
* **TECHNOLOGIES TO BE USED**
* Python
* Tkinter
* API
* **TOOLS TO BE USED**
* Windows 7/8/10/11 versions
* VS Code
* **OVERVIEW**

The currency converter application called Currency Converter, converts amounts from one currency to two others. You can choose to display different currencies like euros, yen, rupees, dollars, etc and enter a value of currency to be converted into the other selected currencies.

* **PRODUCT PERSPECTIVE**

User research and testing: Conducting user research and testing to understand user needs and preferences, and to validate product features and functionality.

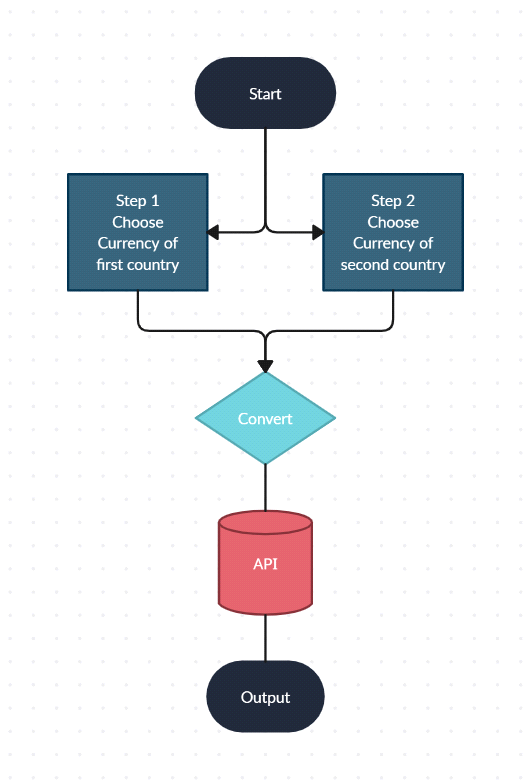
* **INTERFACE**
* **GUI:** Outputs the weather report using tkinter.
* **SYSTEM FUNCTIONS**

A real-time currency exchange rates application is a software solution designed to provide up-to-date and accurate currency information of different countries to users. This type of application is essential for individuals and businesses who need to make decisions based on current currency rates**.**

* **USER CHARACTERISTICS**
* **Exchange rates:** To provide currency exchange rates information specific to the user's interests.
* **Price Tracking**: It provides real-time exchange rates as currency rates keep on fluctuating every moment.
* **Real-time information:** Currency converters aim to maintain real-time information on current market or bank exchange rates, so that the calculated result changes whenever the value of either of the component currencies does.
* **Businesses:** Currency converters help International import and export businesses by helping them determine the selling and buying profits of different products.It allows businesses to differentiate between visitors who are interested in their products and those who are not.
* **Foreign Currency:** An easily accessible online currency converter is very useful to show travelers how their own currencies will fare when exchanged with other foreign currency.
* **FUTURE SCOPE**
* **Incorporating machine learning algorithms:** Predict future exchange rates and provide more accurate conversion rates.
* **Integration with other financial services:** could be integrated with online payment platforms or investment apps.
* **Adding more currencies and crypto currencies:** to include more currencies and crypto currencies.
* **Automating conversion process:** Automatically converting currencies in online shopping carts or in international wire transfers.
* **Implementing a mobile application and Offering a subscription service:** A subscription service with premium features such as real-time exchange rate alerts or historical exchange rate data.
* **Use of block chain technology:** The project could be integrated with block chain technology to allow for decentralized and secure currency exchange.
* **More functionality for businesses:** The ability to generate invoices in different currencies or track expenses in multiple currencies.

**2.1 UML DIAGRAMS**

**Activity Diagram**



**2.2 SOURCE CODE:**

import requests

from tkinter import \*

import tkinter as tk

from tkinter import ttk

class RealTimeCurrencyConverter():

    def \_\_init\_\_(self,url):

            self.data = requests.get(url).json()

            self.currencies = self.data['rates']

    def convert(self, from\_currency, to\_currency, amount):

        initial\_amount = amount

        if from\_currency != 'USD' :

            amount = amount / self.currencies[from\_currency]

        # limiting the precision to 4 decimal places

        amount = round(amount \* self.currencies[to\_currency], 4)

        return amount

class App(tk.Tk):

    def \_\_init\_\_(self, converter):

        tk.Tk.\_\_init\_\_(self)

        self.title = 'Currency Converter'

        self.currency\_converter = converter

        self.geometry("500x250")

        # Label

        self.intro\_label = Label(self, text = '           Currency Convertor          ',  fg = 'blue', relief = tk.RAISED, borderwidth = 3)

        self.intro\_label.config(font = ('Courier',15,'bold'))

        self.date\_label = Label(self, text = f"1 Indian Rupee equals = {self.currency\_converter.convert('INR','USD',1)} USD \n Date : {self.currency\_converter.data['date']}", relief = tk.GROOVE, borderwidth = 5)

        self.intro\_label.place(x = 10 , y = 5)

        self.date\_label.place(x = 160, y= 50)

        # Entry box

        valid = (self.register(self.restrictNumberOnly), '%d', '%P')

        self.amount\_field = Entry(self,bd = 3, relief = tk.RIDGE, justify = tk.CENTER,validate='key', validatecommand=valid)

        self.converted\_amount\_field\_label = Label(self, text = '', fg = 'black', bg = 'white', relief = tk.RIDGE, justify = tk.CENTER, width = 17, borderwidth = 3)

        # dropdown

        self.from\_currency\_variable = StringVar(self)

        self.from\_currency\_variable.set("USD") # default value

        self.to\_currency\_variable = StringVar(self)

        self.to\_currency\_variable.set("INR") # default value

        font = ("Courier", 12, "bold")

        self.option\_add('\*TCombobox\*Listbox.font', font)

        self.from\_currency\_dropdown = ttk.Combobox(self, textvariable=self.from\_currency\_variable,values=list(self.currency\_converter.currencies.keys()), font = font, state = 'readonly', width = 12, justify = tk.CENTER)

        self.to\_currency\_dropdown = ttk.Combobox(self, textvariable=self.to\_currency\_variable,values=list(self.currency\_converter.currencies.keys()), font = font, state = 'readonly', width = 12, justify = tk.CENTER)

        # placing

        self.from\_currency\_dropdown.place(x = 30, y= 120)

        self.amount\_field.place(x = 36, y = 150)

        self.to\_currency\_dropdown.place(x = 340, y= 120)

        self.converted\_amount\_field\_label.place(x = 346, y = 150)

        # Convert button

        self.convert\_button = Button(self, text = "Convert", fg = "black", command = self.perform)

        self.convert\_button.config(font=('Courier', 10, 'bold'))

        self.convert\_button.place(x = 225, y = 135)

    def perform(self):

        amount = float(self.amount\_field.get())

        from\_curr = self.from\_currency\_variable.get()

        to\_curr = self.to\_currency\_variable.get()

        converted\_amount = self.currency\_converter.convert(from\_curr,to\_curr,amount)

        converted\_amount = round(converted\_amount, 2)

        self.converted\_amount\_field\_label.config(text = str(converted\_amount))

    def restrictNumberOnly(self, action, string):

        regex = re.compile(r"[0-9,]\*?(\.)?[0-9,]\*$")

        result = regex.match(string)

        return (string == "" or (string.count('.') <= 1 and result is not None))

if \_\_name\_\_ == '\_\_main\_\_':

    url = '<https://api.exchangerate-api.com/v4/latest/USD>'

    converter = RealTimeCurrencyConverter(url)

    App(converter)

    mainloop()

**Output:**

