Project Plan Group 1

Currency Conversion Application

UMGC CMSC 495 Section 6380

Elizabeth Bloss

Jackson Perry

Carl Blocker

Jonah Kiplimo

Table of Contents

Overview……………………………………………………………………………………………………2

Project Deliverables………………………………………………………………………………………...2

Project Scope………………………………………………………………………………………………..2

Resource List………………………………………………………………………………………………..3

Team Member Roles………………………………………………………………………………………..3

Requirements……………………………………………………………………………………………….4

Case Scenario……………………….………………………………………………………………………6

Schedule Summary.………………..……………………………………………………………………….7

Detailed Schedule….……………………………………………………………………………………….8

Schedule Update & Tracking………………………………………………………………...……………..9

Communication Plan……………………………………………………………………………………….9

Risk Management…………………………………………………………………………………………10

OVERVIEW

This project plan will provide a description of group one’s currency conversion web application. It will contain detailed aspects of the project including deliverables, project scope, team member roles, requirements, case scenarios, and a project schedule. A communication plan will also be included to identify the methods in which information would be shared with stakeholders in a timely manner.

PROJECT DELIVERABLES

The currency conversion web application uses a GUI for users to convert a monetary amount. Users will input a numeric value, select a base currency, and select a conversion currency. After the desired values are input, a ‘convert’ button may be pressed and the converted numeric value will be displayed as output. The project will be delivered through a web application in a container.

PROJECT SCOPE

**Project Scope Statement**

Develop a currency conversion web application where the user will enter information (initial value, base currency, desired conversion currency) and the converted monetary value will be displayed.

**Project Meeting**

The project meetings are scheduled for each Monday at 8pm from 5/21 to 7/8. During the first meeting, topics such as project design, team roles, and project goals were discussed.

RESOURCE LIST

Several resources will be utilized by the team to accomplish the task of creating the website:

1. Personal computers for programming linux/mac or Windows with Docker/WSL
2. IDE: Vscode
3. Python 3.12 and venv
4. Version-controlled software (GitHub repository)
5. Open source API for conversion rates ( openexchangerates.org)
6. Internet access (LAN/Wi-Fi is needed)
7. Libraries outlined in requirements.txt
8. blinker==1.9.0
9. certifi==2025.4.26
10. charset-normalizer==3.4.2
11. click==8.2.1
12. Flask==3.1.1
13. gunicorn==23.0.0
14. idna==3.10
15. itsdangerous==2.2.0
16. Jinja2==3.1.6
17. MarkupSafe==3.0.2
18. packaging==25.0
19. python-dotenv==1.1.0
20. requests==2.32.3
21. urllib3==2.4.0
22. Werkzeug==3.1.3

TEAM MEMBER ROLES

**Project Manager**

This person is responsible for managing the team and organizing meetings. They are responsible for contacting the client about any updates or issues that occur throughout the project. They ensure that all deadlines are met and requirements are completed. The project manager for group one is Jackson Perry.

**Programmer**

All team members can work together to ensure the function and logic of the tools algorithms as well as ensuring performance and safety of the application. This is to ensure that integration testing is performed as development happens and acceptance testing is thoroughly completed prior to delivery with a focus on security and accuracy.

**User Interface Designer**

These individuals are responsible for designing and layout of the graphical user interface for the project. The frontend of the program will be developed with HTML/CSS/tailwinds The team as a group will manage this portion of the project.

**Tester**

All team members will serve as testers and complete a rigorous test plan as outlines by the Project manager. Sufficient assertion within the codebase is also desired. We will test for functionality and correctness but also for appropriate error handling and graceful failure .

REQUIREMENTS

**Business Requirements**

1. Allow users to convert one currency to another in near real-time.

* Support popular global currencies (USD, EUR, GBP, INR, JPY, etc.).
* Display accurate and up-to-date exchange rates.
* Ensure easy usability on both desktop and mobile devices.
* Provide a secure and fast user experience to retain users.

**Functional Requirements**

* **Currency Selection**: User selects source and target currencies from dropdowns.
* **Input Field**: User inputs the amount to convert.
* **Conversion Calculation**: App retrieves current exchange rates and calculates the converted value.
* **Display Result**: Show converted amount clearly to the user.
* **Exchange Rate Source**: Integration with a reliable external API (e.g., [OpenExchangeRates](https://openexchangerates.org/)).
* **Auto-Refresh Rates**: Periodically update exchange rates (e.g., every 10 mins or on demand).
* **Currency Symbol and Code Display**: Show symbols ($, €, ₹) and ISO codes (USD, EUR, INR).
* **Mobile Responsiveness**: Ensure UI adapts well to various screen sizes.
* **Error Handling**: Display error messages for invalid inputs or failed API calls.
* **History (Optional)**: Maintain a list of recent conversions.
* **Localization (Optional)**: Support multiple languages or regional formats.

**Technical Requirements**

* **Frontend**:
  + Language: HTML, CSS
  + /Tailwind
* **Backend** (if applicable):
  + Language: Python (Flask)
  + Use server-side code to call external currency exchange API
* **API Integration**:
  + Must fetch exchange rates from a real-time API.
  + Handle rate limits and API key security
* **Performance**:
  + Fast load times (<2s on average internet)
  + Minimized API calls (cache responses where possible)
* **Security**:
  + Secure API keys (never expose them in frontend)
  + HTTPS for all data transmission
  + Input validation to prevent injection or malicious input
* **Hosting/Deployment**:
  + Use GitHub for version control
* **Browser Compatibility**:
  + Should work across major browsers (Chrome, Firefox, Safari, Edge)
* **Hardware**
  + X86 processor with at least 4 threads

CASE SCENARIO

**User Journey:**

**1. Launching the Application**

* + The user opens the app and lands on the Welcome Screen.
  + The screen displays a main menu with the following options:
  + Currency Converter
  + Conversion History
  + Exit

**2. Currency Conversion**

* + The user selects the "Currency Converter" button.
  + The Currency Converter screen opens.
  + They are presented with a clean Tailwind-styled form that includes:
  + Input field to enter the amount (e.g., 100)
  + Dropdown to select source currency (e.g., USD)
  + Dropdown to select target currency (e.g., EUR)
  + A “Convert” button
  + The user enters the value 100, selects USD as the source and EUR as the target, and clicks "Convert".
  + The converted amount appears below the form:
  + 100 USD = 92.34 EUR (check via Api for correct rate at test time)
  + The conversion is saved in the backend session as history.
  + The user clicks "Exit" to return to the main menu.

**3. Conversion History**

* + From the main menu, the user selects "Conversion History".
  + A panel opens showing the last three conversions made in the current session:
    - 100 USD → 92.34 EUR
    - 50 GBP → 63.12 USD
    - 200 JPY → 1.30 EUR
  + The user can click "Print History" to export the session log to PDF or print using a local printer.
  + The user returns to the main menu and selects "Exit" to close the app.

**4. Error Handling Scenarios:**

* + Invalid Input – Amount
  + User enters a non-numeric amount (e.g., "ten")
  + The system shows a Tailwind-styled error banner:
  + Please enter a valid numeric amount.
  + Invalid Currency Selection
  + User enters a manually typed currency code not in the dropdown (e.g., XYZ).
  + Error message appears:
  + XYZ' is not a supported currency code.
  + Empty Field Submission
  + User clicks "Convert" without filling all fields.
  + Red outlines appear on missing fields with a message:
  + All fields are required to perform the conversion.
  + API Failure / No Internet
  + If the exchange rate API is down or there’s no internet:
  + Unable to retrieve exchange rates. Please try again later.

**Success Criteria:**

* User can complete at least 3 conversions in one session.
* Conversions are saved temporarily in session memory.
* Tailwind ensures responsive layout on all devices.
* User receives clear error messages for all input errors.

SCHEDULE SUMMARY

**Currency Conversion Application – Development Schedule with Milestones**

**(CMSC 495: Group 1 Project, Summer 2025)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **May** | **June** | **July** |

Detail Target End

W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Meet Team** | 5/21/25 |  |  |  |  |  |  |  |  |
| **Project Plan** | 5/27/25 |  |  |  |  |  |  |  |  |
| **Project Design** | 6/3/25 |  |  |  |  |  |  |  |  |
| **Phase I Build** | 6/10/25 |  |  |  |  |  |  |  |  |
| **Test Plan** | 6/17/25 |  |  |  |  |  |  |  |  |
| **Phase II Build** | 6/24/25 |  |  |  |  |  |  |  |  |
| **User Guide** | 7/1/25 |  |  |  |  |  |  |  |  |
| **Final Report** | 7/8/25 |  |  |  |  |  |  |  |  |

DETAILED SCHEDULE

**Target Start Target End**

**Meet Team:**  05/14/25 05/21/25

Discuss Roles 05/21/25

Communication Plans 05/21/25

Discuss Project Options 05/21/25

**Project Plan:** 05/21/25 05/27/25

Allocate Roles 05/27/25

Discuss Requirements 05/24/25

**Project Design:** 05/27/25 06/03/25

Config File Load 06/03/25

Backend coding 06/02/25

Frontend coding 06/02/25

**Phase I Build:** 05/30/25 06/10/25

Config File Load 06/09/25

Backend development 06/09/25

**Test Plan:** 06/10/25 06/17/25

Indiv Test Plans 06/15/25

First Draft (Test Plans) 06/12/25

Final Draft (Test Plans) 06/16/25

**Phase II Build:** 06/03/25 06/24/25

User UI 06/23/25

**User Guide:** 06/25/25 07/01/25

Indiv User Guides 06/30/25

First Draft (User Guide) 06/27/25

Final Draft (User Guide) 06/30/25

**Final Report:** 06/17/25 07/08/25

Compile Report 07/07/25

Presentation 07/07/25

SCHEDULE UPDATE & TRACKING

**Currency Conversion Application – Development Schedule with Milestones**

**(CMSC 495: Group 1 Project, Summer 2025)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **May** | **June** | **July** |

Detail Target End

W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Meet Team** | 5/21/25 | COMPLETE |  |  |  |  |  |  |  |
| **Project Plan** | 5/27/25 |  |  |  |  |  |  |  |  |
| **Project Design** | 6/3/25 |  |  |  |  |  |  |  |  |
| **Phase I Build** | 6/10/25 |  |  |  |  |  |  |  |  |
| **Test Plan** | 6/17/25 |  |  |  |  |  |  |  |  |
| **Phase II Build** | 6/24/25 |  |  |  |  |  |  |  |  |
| **User Guide** | 7/1/25 |  |  |  |  |  |  |  |  |
| **Final Report** | 7/8/25 |  |  |  |  |  |  |  |  |

COMMUNICATION PLAN

Team members will communicate via the LEO “Group 1 Members Only” discussion group and daily through Discord. Group members will also manage version control via a GitHub repository “CMSC495” created by Jackson Perry.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **CONTACT LIST** |  |  |
| **Name** | **Org** | **Role** | **Discord** | **GitHub** |
| Elizabeth Bloss | UMGC  CMSC 495 | Programmer | E\_Bloss | erbloss |
| Jackson Perry | UMGC  CMSC 495 | Project Manager/ Programmer | jackson1982\_ | jackson-perry |
| Carl Blocker | UMGC  CMSC 495 | Programmer | Cblock11 | Cblock11 |
| Jonah Kiplimo | UMGC  CMSC 495 | Programmer | lymore\_afk | lymore-afk |
| Hung Dao | UMGC  CMSC 495 | Instructor | N/A | N/A |

RISK MANAGEMENT

There can be several risk factors that may result in project delays or that could affect the overall efficiency of the program. Some of the risks that can be anticipated are:

* Team Availability
  + We may fall behind due to scheduling conflicts if we are unable to consistently meet and discuss the development of the project
  + To resolve this issue, we will predetermine a set time for weekly communication where we will discuss what we have accomplished and what needs to be done. There will also be an ongoing chat forum via Discord
* Teamwork
  + Team collaboration in computer science is a new territory for some in the group. Members will learn to work together in a virtually remote setting and use version control software (GitHub) to submit a cohesive program.
* Unfamiliarity with specific programming languages
  + There are some members that may be more familiar with the languages chosen to create the program. Each of us is expected to familiarize ourselves with the languages that we will be using so that we are better prepared to contribute to the group. Open and respectful communication among members will enable us all to grow as programmers.