## Exercise 4: Build an Interactive Interest Calculation App Using Claude

## Objective

You will use Claude to design and build an interactive Interest Calculation App that performs:

* Simple Interest calculations
* Compound Interest calculations
* Multi-year growth projections
* Optional CSV/Excel upload of financial inputs
* Download of computed results

You will prompt Claude to help design, structure, and implement the app using Claude Artifacts, ensuring your code and interface remain organised and reusable.

## Example Financial Fields

Your interest calculator may include the following inputs:

1. Principal Amount (GBP)
2. Interest Rate (%)
3. Term Length (Years)
4. Compounding Frequency

(Annual, Monthly, Daily, Continuous)

1. Final Value
2. Total Interest Earned
3. Year-by-Year Breakdown (Optional)

# Example Prompts to Use with Claude

### Prompt 1 – Designing the App

“Help me design an interactive Interest Calculator App.

It should support Simple Interest, Compound Interest, and yearly breakdowns.

Suggest the structure, fields, and UI layout for an easy-to-use calculator.”

### Prompt 2 – Choosing the Technology & Using Artifacts

“Explain how Claude’s Artifacts feature can help me build the calculator.

What files, components, or modules should I create?

Recommend whether to build the app as HTML/JS, a Python script, or another format.”

### Prompt 3 – Core Logic for Interest Calculations

\*“Write the core calculation logic for:

1. Simple Interest
2. Compound Interest (configurable compounding frequency)
3. A function that produces a yearly projection table.”

A black background with white text

AI-generated content may be incorrect.

### Prompt 4 – Building the Interactive App

“Create an interactive version of the Interest Calculator using an Artifact.

Include form inputs, a calculate button, and output results.

Use HTML+JavaScript or Python (choose whichever structure you recommended earlier).”

### Prompt 5 – Adding File Upload Capability (Optional / Advanced)

“Add support for uploading a CSV or Excel file containing principal amounts, interest rates, and terms.

Process the file, calculate interest for each row, and provide a downloadable results file.”

### Prompt 6 – Enhancing User Experience

“Suggest optional enhancements such as charts, sliders for interest rate, currency formatting, or toggles between yearly and monthly views.”

### Things to Consider

Students should reflect on the following design choices:

1. User Flow
   * Input values → Calculate → Display Results → Export Output
2. Input Validation
   * Ensuring numbers are valid
   * Handling missing or zero values
   * Preventing invalid compounding frequencies
3. Modularity
   * How Artifacts help separate UI, logic, and data
4. Extensibility
   * Could the same structure later support:
     + Mortgage calculators?
     + Investment projections?
     + Savings goals?
5. User Experience
   * Should results include:
     + A graph?
     + A summary card?
     + A downloadable report?

# Key Skills Practiced

### Technical Skills

1. Structured prompting to get Claude to assist in building real applications
2. Planning and modelling financial calculators
3. Separating logic, interface, and data into clean components
4. Building interactive tools using Claude Artifacts
5. Creating file-upload and file-download pipelines
6. Implementing common finance formulas in code
7. Designing user-friendly UI workflows
8. Understanding how to iterate and improve apps using AI support

### Soft Skills & Thinking Skills

1. Breaking down complex problems into small, solvable prompts
2. Applying financial reasoning to application development
3. Evaluating different implementation approaches (web, Python, etc.)
4. Considering real-world use cases for interest calculators