# 3 Web Projects Assignment

**Total Marks: 40**

---

## Project 1 — Calculator with Currency Converter (Total: 40 marks)

### Overview

UI mein **2 tabs** hongay: `Calculator` aur `Currency Converter`.

### Calculator (Scientific)

- Scientific calculator — **kam az kam 12 operations** include karne hain. (Example operations: addition, subtraction, multiplication, division, power, square root, factorial, sin, cos, tan, ln/log, percent, inverse)
- Input aur result display clearly show hona chahiye.
- **Last 10 calculations** ka history panel dikhayega.
  - Har history item ko **individual clear** karne ka button hoga.
  - **Clear All** button bhi hoga jo poori history remove karega.
- Keyboard se input support achha hoga (optional but recommended).

### Currency Converter

- Do input fields: **Amount** aur **From / To** dono ke liye dropdowns jisme currencies select kar sakay.
- Currency rates **real-time** API se fetch hongi (public currency conversion API use karna — student ko API key/endpoint comment mein mention karna chahiye).
- Conversion result turant show hoga.
- **Last 10 conversions** ka history yahan bhi store hogi, jisko individual clear aur clear all ki option hogi.
- Error handling: agar API fail kare to user-friendly message show karein.

### UI/UX Requirements

- Bootstrap-based responsive layout.
- Tabs clear hon, mobile friendly design.
- History list scrollable aur readable.

### Deliverables

- `index.html`, `styles.css`, `script.js` (or organized structure) + brief README jisme API used ka naam aur koi setup step likha ho.

### Marks Distribution (15)

- Functionality (calculator operations + currency conversion): **10**
- History (working last 10, individual clear, clear all): **10**
- UI & Responsiveness: **5**
- Error handling & API use: **10**
- Code quality: **5**

---

## Project 2 — Dynamic Flip-Card Matching Game (Total: 40 marks)

### Overview

Simple memory matching game jisme user pehle **3 inputs** dega:

1. **Name** (player name)
2. **Layout** (perfect square number — example: 9, 16, 25, 36, 100)
3. **Time** (minutes for the game timer)

**Layout** ko perfect square lena zaroori hai taake grid dynamic ban sake: `n = sqrt(layout)` rows aur columns.

### Gameplay

- Game start karne par cards grid mein show honge (face-down).
- Jab user ek card pe click karega to card flip hoga; doosre card pe click karne par woh bhi flip hoga.
- Agar dono cards **match** karte hain to point increase hoga aur wo cards remove/disable ho jayenge.
- Agar match nahi to dono cards wapas face-down flip ho jayenge after short delay (e.g., 600ms).
- Timer chal raha hoga — jo time user ne dala tha (minutes). Agar timer khatam ho gaya toh game over.
- Score display aur remaining pairs count show karein.

- Optional: best score / localStorage mein save karna (extra credit).

### UI/UX Requirements

- Grid responsive; cards square-shaped.
- Flip animation CSS-based (no external heavy libs required).
- Start / Restart / Pause buttons.

### Deliverables

- `index.html`, `styles.css`, `script.js` + README with instructions.

### Marks Distribution (15)

- Dynamic grid + correct perfect-square handling: **10**
- Game logic (flip, match, scoring): **10**
- Timer & game over handling: **10**
- UI/animations & responsiveness: **5**
- Code quality: **5**

---

# Project 3 — Google-Form-like Builder (Total: 40 marks)

### Overview

Ek form builder jaisa Google Forms:

### Features

1. **First section**: 2 inputs — **Title** aur **Description** (editable).
2. **Add Questions**: User jitne questions chahay add kar sakta hai.
3. **Question Types** (each question must allow selecting type):
   - Short answer
   - Long paragraph
   - Multiple choice
   - Checkboxes
4. **Options management**:
   - Agar question type `Multiple choice` ya `Checkboxes` ho to options add karne ka UI ho.
   - Har option ko add / delete kar sakay.

5. **Question actions**:
   - Delete question
   - Duplicate question (same type + options copied)
6. Preview mode: Form ko preview karke answers type kar sakay (optional but useful).

### UI/UX Requirements

- Drag & drop ordering is optional (extra credit). Basic up/down move buttons acceptable.
- Clean Bootstrap form layout.

### Deliverables

- `index.html`, `styles.css`, `script.js` + README.

### Marks Distribution (10)

- Add/delete/duplicate question functionality: **10**
- Options add/delete per question + type switching: **10**
- Title/description + preview: **10**
- UI & Code quality: **10**

---

# General Notes

- **Tech stack**: HTML, CSS (Bootstrap), JavaScript (vanilla). No frameworks required.
- Each project should have a small `README.md` explaining how to run locally and any API keys/config needed.
- Submit zipped folder with clear structure.
- Clear code comments expected.