🧠 Core Logics Implemented

◇ 1. **Campus-Based Shortcut Logic**

- **If the program is located at *Obuasi Campus*:**
  - ☑ Automatically qualified.
  - ⚙ No need to check main/tracks/electives.
  - ➕ All unique 3-subject combinations from student electives are considered valid.

◇ 2. **Student Elective Input Handling**

- Student electives are passed as:

```
["Subject1", "Subject2", "Subject3", ...]
```

◇ 3. **Requirement Structure Parsing**

- `program["elective subjects"]` may include:
  - `"main"` → required subject(s) (can be a single string or list).
  - `"tracks"` → disciplines with grouped elective subject conditions.
    - Each track is a list that may include:
      - Single subjects
      - Lists → Mutually exclusive group (select **one**)

◇ 4. **Track Qualification Logic**

For each track under a program:

- ☑ Student qualifies **if all required parts** of the track are matched:
  - For each element in the track:
    - If it's a string: it must exist in student electives.
    - If it's a list (mutually exclusive group): at least one item in the list must exist in student electives.
- A track qualifies if:
  - **All track conditions** are satisfied.
  - ☑ Only **one** track match is needed for program qualification.

◇ 5. **Main Subject Validation (if applicable)**

- If `"main"` is defined:
  - It must exist in the student electives.
  - `main` can be a string or list (any match from list qualifies).

◇ 6. **Deduplication of Electives**

- Duplicates in electives are removed (e.g., `"Accounting", "Accounting"` → `"Accounting"` once).

◇ 7. **Combination Generation**

- All **3-subject combinations** are generated from the validated student electives.
- Each combination is:
  - Only included if it contains the **main subject** (if defined).
  - Always included if program is from **Obuasi Campus**.
- Duplicates are removed using `set` and then converted back to lists.

◇ 8. **Validation Flow Summary**

For each program:

1. If campus is Obuasi → **auto-qualified**
2. Else:
   - Validate against all tracks:
     - Check `main` (if any)
     - Check all required elements in any one track
   - If qualified → generate all valid 3-subject combinations

---

◇ 9. **Result Formatting**

- Final output for each program includes:
  - Program name
  - Campus
  - Whether student is **qualified**
  - List of all **valid 3-subject combinations**

Up Next

Check program requirements for business. Different subject names with the subject requirements.