Answers & Observations

- **Question 1: How Many Test Scenarios Are Necessary?**

  A total of 25-30 test scenarios are necessary to verify the calculator modal works as expected, distributed as follows:

  UI Test Scenarios: 15 scenarios
  Core Functionality:

  - Does the modal open automatically with correct default values?
  - When I change the amount, does the monthly payment update?
  - When I change the period, does the payment recalculate?
  - If I click JÄTKA, do my changes save to the header?
  - If I close without JÄTKA, do changes get discarded?
  - Can I reopen the modal and see my saved values?

  Boundary & Validation:

  - What happens at minimum values (500€, 6 months)?
  - What happens at maximum values (30,000€, 120 months)?
  - Can users enter negative numbers or letters?
  - Does the UI prevent obviously invalid inputs?

  API Test Scenarios: 15 scenarios
  Happy Path:

  - Does the API return valid calculations for standard loans?
  - Is the monthly payment reasonable (positive, less than loan amount)?
  - Are results consistent when I send the same request twice?

  Business Logic:

  - If I double the amount, does the payment roughly double?
  - If I extend the period, does the monthly payment decrease?
  - Is APRC higher than the base interest rate (proving fees are included)?

  Edge Cases:

  - Does the API accept minimum values (500€, 6 months)?
  - Does it reject amounts below 500 or above 30,000?
  - What happens with negative numbers or missing fields?

- **Question 2: API Testing Approach for Monthly Payment & APRC**

  - Verifies payment a positive number?
  - Verifies the Monthly Payment?
  - Is it more than just principal divided by months (proving interest is included)?
  - Same inputs always produce same outputs (no randomness)
  - Does the same loan always return the same APRC?

- **Question 3: What Should Be Automated?**

  - I'll automate the critical, and repetitive scenarios which includes:
    1) All API Tests – because it catches the bugs early
    2) Boundary value cases
    3) Recalculation of Input Changes

- **Observations**
  - UI and API validation mismatch for Loan Period

    The UI does not allow decimal values for loan period (e.g. 60.5), whereas API Allows.
  - UI and API validation mismatch for maximum and minimum Loan and Period values

    UI does allow to input less than 500 eur, whereas API accepts amounts lower than the UI minimum without error. Same for higher amount.
  - Inconsistent Error Handling When Fields Are Missing

    Response showing when Interest rate field is not accurate, also no error message showing when Currency Field is unavailable.
  - UI allows entering an amount value greater than the maximum allowed limit

    Despite the maximum allowed amount being 30,000, the UI allows users to enter higher values (e.g., 1,000,000) without restriction.