Project Workflow Recap (for every answer, as requested)

- **Commit and push changes:**
  `git add .` → `git commit -m "message"` → `git push`
- **Build and deploy site:**
  `npm run build` → `npm run deploy`
- **GitHub Pages settings:**
  Should be set to `gh-pages` branch, `/ (root)` folder.
- **See your live app:**
  https://ca-1969.github.io/ADVANCE-TAX-INTEREST-CALCULATOR/

Review: Are Your Interest Calculations Correct for 234A, 234B, 234C for FY 2024-25/AY 2025-26?

Section 234A (Delay in Filing the Return)

- Standard: **1% per month or part thereof on the unpaid tax** from the due date (usually July 31) to actual return filing date.
- **Your React formula:**

  ```
  interest234A = monthsDelay > 0 ? taxPayable * 0.01 * monthsDelay : 0;
  ```

  - `monthsDelay` is calculated as months between return due date and actual filing date + 1.
  - **Correct** for most assessment scenarios (monthly compounding on whole or part of month per Finance Act 2025).

Section 234B (Default in Payment of Advance Tax)

- Standard: **1% per month from April 1 to assessment date on unpaid advance tax shortfall** (generally if paid <90% of total tax).
- **Your formula:**

  ```
  if (taxLiability > 0) {
    const shortfall = taxLiability;
    if (shortfall > 0.1 * (taxLiability + advanceTaxPaid)) {
      const months = ...;
      interest234B = shortfall * 0.01 * months;
    }
  }
  ```

    - This calculates interest on the shortfall (difference between required and paid), and checks if the shortfall is >10% of total liability (correct per Act).
    - The period calculation uses assessment year logic, but `currentDate.getMonth() - 3` may sometimes incorrectly compute the number of months since April 1.
        - This is because JavaScript months are zero-based (January=0). For correct calculation, consider:

          ```
          const months = Math.max(
            0,
            (currentDate.getFullYear() - 2026) * 12 +
            (currentDate.getMonth() + 1 - 4) // April is 4th month
          );
          ```

        - If assessment year changes, this needs update.

Section 234C (Deferment of Advance Tax)

- Standard: **Interest for deferment/shortfall in quarterly installments as specified under section 234C**
    - 1% per month (fixed number of months for each period) on shortfalls.
- **Your formula:**

  ```
  for (let i = 0; i < 4; i++) {
    const required = (taxLiability + advanceTaxPaid) * duePercents[i];
    const paid = ...;
    if (currentDate > due && paid < required) {
      interest = (required - paid) * 0.01 * period; // period: [3,3,3,1]
      interest234C += interest;
    }
  }
  ```

    - This logic is **functionally correct**: checks if the deadline has passed and the installment was underpaid, and applies the statutory period for calculation.

Summary:

- **Your app currently calculates 234A, 234B, 234C interest using correct standard methods for FY 2024-25 (AY 2025-26) as per Finance Act 2025**.
- **Minor caution:** For 234B, the month calculation should be double-checked for cross-year assessment or for assessment year increment in the future. Right now, it works if `currentDate` is always within the correct year.

For Future Years: Keeping Calculator Updated for Each Finance Act

To allow users to calculate for different years:

1. Add a **dropdown menu** to select Assessment Year (AY).
2. Use a data structure to store any year-specific rates, due dates, and logic.

3. On selection, use the appropriate constants/formulas for that year.
4. Retain old logic for previous years, and add new logic per each Finance Act when published.

Example structure:

```
const FINANCE_ACTS = {
  "2025-26": {
    dueDate: "2026-07-31",
    sectionRates: { a: 0.01, b: 0.01, c: 0.01 },
    // ...other parameters
  },
  // For future years, add/modify as needed
  "2026-27": { ... }
}
```

- On UI, allow user selection, e.g., “Select Year:”.
- Use selected year’s parameters in calculations.

Google Sheets Reminder (Always):

- Row 1: Paste and bold these headers:

  ```
  Date | Name | Mobile Number | Email ID | Return Filing Date | Tax Payable | Advance T
  ```

- Select Row 1 → Click **B** to bold.

Project Update/Deploy Workflow (Repeat):

- `git add .` → `git commit -m "message"` → `git push`
- `npm run build` → `npm run deploy`
- Check your GitHub Pages Settings (**Branch:** `gh-pages`**, Folder:** `/ (root)`)
- Test site: https://ca-1969.github.io/ADVANCE-TAX-INTEREST-CALCULATOR/

Conclusion:
Your formulas are correct for FY 2024-25, and the structure is easy to extend for FY 2025-26/2026-27.
Let me know if you want *immediate* sample code for the drop-down year selection and changeover logic!