Common Return Equations for All Funds

NAV Plus (`mf_nav`) · Database `fintra_mf` · Prepared by Shafiul Mahmud Partho · 26 October 2026 · Data up to 25 Oct 2026 (IDLC up to 19 Oct 2026)

> Every fund gets one series, **K(t)**: a composite NAV with dividends reinvested. Return growth, since-launch return, CAGR, fund age, day change and calculator projections are all read off K(t) with the same equations. Only the way K(t) is built differs, in four cases.

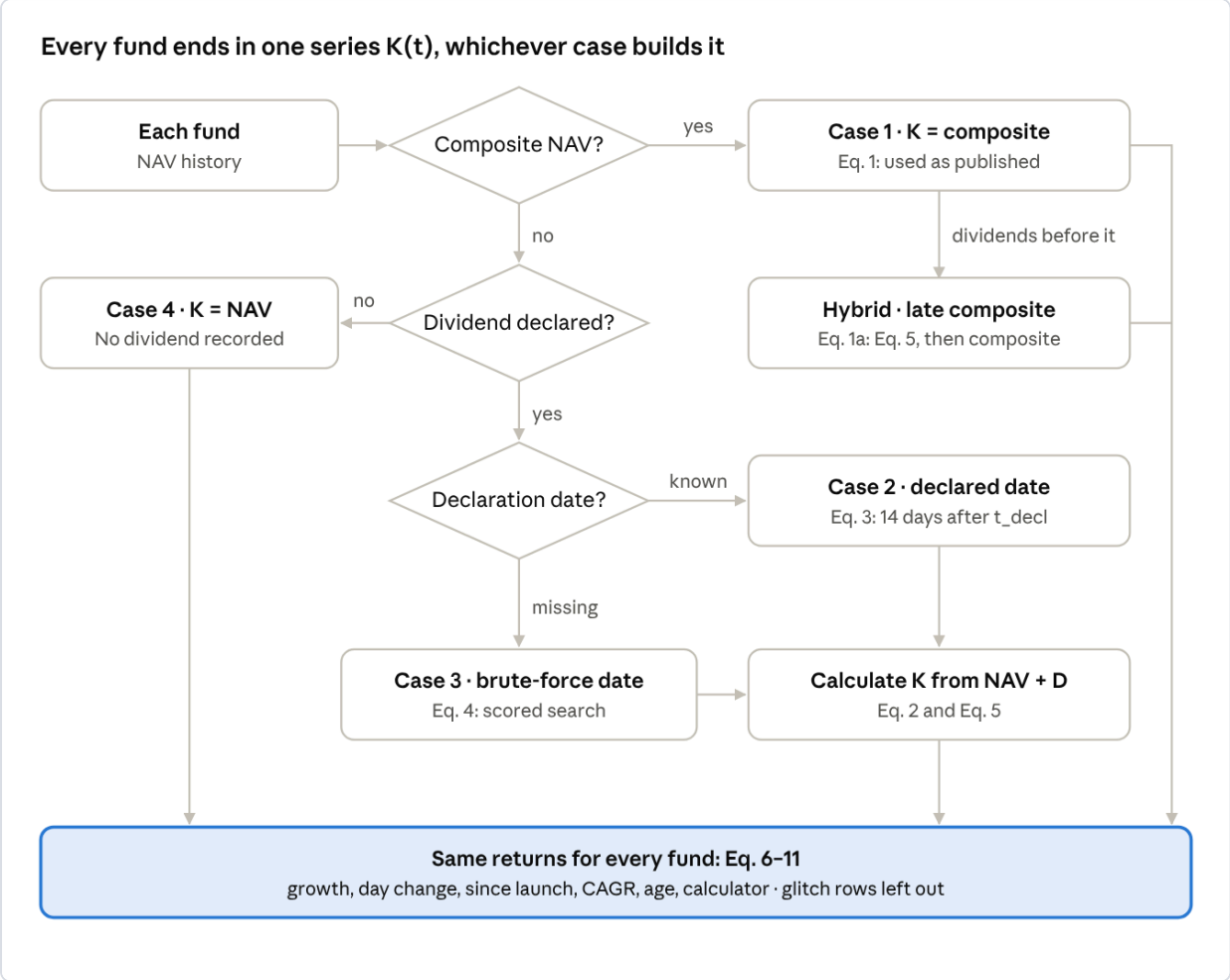
1. Summary
2. Decision flow
3. Inputs and database mapping
4. The equations
5. Pseudocode
6. Worked examples from the database
7. Brute-force validation
8. Data to fix before rollout
9. What changes in the project

1. Summary

Case	Condition	How K(t) is built	Funds today
1	The AMC publishes a composite NAV	Taken from `composite_nav`, exactly as the AMC publishes it	16 (EDGE, Ekush, Shanta, VIPB)
Hybrid (1a)	A composite NAV exists, but dated dividends fall more than 31 days before it first differs from NAV	Calculated as Case 2 or 3 until the composite starts, then follows the composite's daily moves	0
2	No composite NAV; dividend declared with its declaration date	Built from NAV + dividend per unit, placed on the effective date found after the declaration date	1 (CWT Sadharan Bima, whose 2 rows are suspect)
3	No composite NAV; dividend declared but its date is missing	Effective date found by a brute-force search, then built as Case 2	1 (IDLC Income)
4	No composite NAV; no dividend declared	K(t) = NAV	9 (CWT, IDLC, UCB)

The rule is simple: a published composite NAV is used as it is; only when none exists is it calculated from NAV and dividends. The brute-force search found the exact effective date for **14 of 14** Shanta dividends when tested without their declaration dates.

2. Decision flow



Each fund takes the first branch that matches. Every case ends in the same series K(t), and every return figure is read off it with the same equations. The case is chosen again on every rebuild, so a fund moves between cases as composites, dividends and dates arrive.

3. Inputs and database mapping

All inputs already exist in `mf_nav_history`, `mf_dividend_history` and `mf_funds`. New columns record which case a fund uses, how each dividend was dated, and the resulting series K(t).

Symbols and where they come from

Symbol	Meaning	Table.column	Status
N(t)	NAV per unit on date t (Tk)	`mf_nav_history.nav`, `as_of_date`	Exists
P(t)	AMC-published composite NAV	`mf_nav_history.composite_nav` (a fund has one when it differs from `nav` on some rows)	Exists
F	Face value per unit, normally Tk 10	`mf_funds.face_value_per_unit` (VARCHAR: parse as a number, default 10)	Exists
p	Declared dividend, % of face value	`mf_dividend_history.dividend_percentage`	Exists
FY	Financial year label	`mf_dividend_history.financial_year` ("2023" or "2025-26")	Exists
t_{decl}	Declaration date	`mf_dividend_history.declaration_date`	Exists; must become NULL when unknown instead of the 30 June placeholder
D	Cash dividend per unit (Tk)	`mf_dividend_history.dividend_per_unit`	New
t_{eff}	Day the dividend leaves the NAV	`mf_dividend_history.effective_date`	New
t_0	First NAV date: the earliest NAV the fund declared (glitch rows skipped)	`mf_funds.series_start_date`, set by the rebuild	New
T	Latest NAV date	`mf_funds.latest_nav_date`	Exists
K(t)	The fund's series: composite NAV with dividends reinvested	`mf_nav_history.adjusted_nav`	New

New columns and table

Table	Column	Type	Meaning
`mf_funds`	`series_basis`	ENUM('COMPOSITE','HYBRID','DIVIDENDS','NAV')	Case 1, Hybrid, Cases 2–3, or Case 4; set by the rebuild
`mf_funds`	`series_start_date`	DATE NULL	t_0, used by Equations 8 to 10
`mf_funds`	`cagr`	DECIMAL(8,2) NULL	Equation 9; NULL while the fund is under a year old

Table	Column	Type	Meaning
`mf_dividend_history`	`dividend_per_unit`	DECIMAL(12,4)	Equation 2
`mf_dividend_history`	`effective_date`	DATE NULL	Equation 3 (Case 2) or Equation 4 (Case 3)
`mf_dividend_history`	`date_source`	ENUM('DECLARED','SEARCH','ADMIN')	How the effective date was found
`mf_dividend_history`	`match_score`	DECIMAL(8,4) NULL	Search score; lower is a closer match
`mf_dividend_history`	`effective_confirmed`	BOOLEAN	Set by an admin; unconfirmed search results are listed for review
`mf_nav_history`	`adjusted_nav`	DECIMAL(16,6)	K(t), rebuilt after each NAV scrape or dividend change. Stored on every row; only rows from t_0 on that are not glitch rows form the series
`mf_nav_history`	`is_glitch`	BOOLEAN	Set by an admin; the row is left out of K(t), the date search and range boundaries
`mf_unit_adjustments`	`fund_id`, `effective_date`, `factor`, `reason`, `source`	New table	Unit restructurings for Cases 2–4 only; a published composite already includes them

`composite_nav` and `investor_return` keep the AMC's published values. `inception_return` is rewritten as the since-launch return of Equation 8.

4. The equations

Equations 1 to 5 (with 1a for a late composite) build K(t); Equations 6 to 11 read every return figure off it.

Case 1: a composite NAV is available

Equation 1. Case 1 series

When the AMC publishes a composite NAV, or one is scraped, it is used exactly as published. No type check and no conversion.

$$K(t) = P(t)$$

Equation 1a. Hybrid series (a composite that starts late)

Some funds may publish a composite only from a later date, after dividends were already paid; before that date the stored composite equals NAV. Let t_c be the first date where P(t) differs from N(t). If a dated dividend falls more than 31 days before t_c, the composite was not published then, so K(t) is calculated up to t_c and follows the composite after it:

$$K(t) = \begin{cases} \text{Equation 5}, & t \le t_c \\ K(t^-) \times \dfrac{P(t)}{P(t^-)}, & t > t_c \end{cases}$$

The step into t_c uses NAV and dividends, so a composite that jumps by its cumulative dividends on its first day is not counted twice. With no dated dividend before t_c, Equation 1 applies unchanged. No fund needs the hybrid today: every composite starts at its fund's first dividend.

Cases 2 and 3: no composite, so K(t) is calculated

Equation 2. Dividend per unit

$$D = \frac{p}{100} \times F \qquad (\text{e.g. } 26\% \times \text{Tk } 10 = \text{Tk } 2.60)$$

Equation 3. Case 2 effective date (declaration date known)

$$t_{eff} = \underset{t \in [\, t_{decl},\, t_{decl}+14 \text{ days}\,]}{\arg\min} \left| \big(N(t^-) - N(t)\big) - D \right|$$

t^- is the NAV date before t. The match counts only when the NAV drop is at least 50% of D. With no such drop in the window, t_{eff} is the first NAV date after t_{decl}, flagged for an admin to confirm.

Equation 4. Case 3 effective date (declaration date missing): brute-force search

Every NAV date in a window is scored and the best one wins. The window is centred on the fund's usual payout timing, learned from its dividends whose dates are known.

$$FY_{end} = 30 \text{ June of the label's last year}, \qquad o = \text{median}_{k \in \text{dated}} \left(t_{eff,k} - FY_{end,k}\right), \qquad E = FY_{end} + o$$

$$\text{score}(t) = \frac{\left|\left(N(t^-) - N(t)\right) - D\right|}{D} + 0.5 \times \frac{|t - E|}{45 \text{ days}}, \qquad t_{eff} = \underset{t \in [\,E-45,\,E+45\,]}{\arg\min} \text{score}(t)$$

- FY_{end} examples: "2023" gives 30 Jun 2023; "2025-26" gives 30 Jun 2026.
- Only dates whose drop is at least 50% of D count. Glitch rows and dates already used by another dividend of the fund are skipped.
- The offset o comes from the fund's own dated dividends; with none, from the AMC's; with none at all, the window widens to FY_{end} − 6 months … FY_{end} + 9 months and the timing term is dropped.
- No candidate means no date: the dividend is listed for an admin and left out of K(t) until dated.
- Every search result is saved with `date_source = SEARCH` and its score, and stays unconfirmed until an admin checks it.

Equation 5. The calculated series (Cases 2, 3 and 4)

$$K(t_0) = N(t_0), \qquad K(t) = K(t^-) \times \frac{s(t) \times N(t) + D(t)}{N(t^-)}$$

D(t) is the dividend per unit on its effective date and 0 on every other date. s(t) is the unit restructuring factor from `mf_unit_adjustments`, 1 on every other date. Case 4 has no dividends, so K(t) = N(t).

Glitch rows. A NAV row an admin flags (`is_glitch`) is dropped before any equation runs: t_0 is the first unflagged NAV row, t^- is the previous unflagged row, and the row still stores an `adjusted_nav` (the previous K moved by the row's own price change) for display only. A one-day jump that reverts therefore never becomes a range start or end, a dividend date, or a CAGR endpoint. Three rows are flagged today: CWT EBFG 10 Apr 2026 (−21%), EGF 31 Mar 2022 (+35%) and ESRF 25 Oct 2026 (composite 56.06 against NAV 15.606).

Common equations: the same for every fund

Equation 6. Return growth for a range (1M, 6M, 1Y, 3Y, 5Y, All)

$$b = \max\{\, t : t \le T - \text{range} \,\}, \qquad G(t) = \left(\frac{K(t)}{K(b)} - 1\right) \times 100, \qquad R = G(T)$$

Ranges are calendar months and years. The chart runs from b to T, so its last point always equals the headline R. A fund younger than the range uses b = t_0, labelled "Since launch".

Equation 7. Day change

$$\Delta_{day} = \left(\frac{K(T)}{K(T^-)} - 1\right) \times 100$$

Equation 8. Since-launch return (stored as `inception_return`)

$$R_{launch} = \left(\frac{K(T)}{K(t_0)} - 1\right) \times 100$$

Equation 9. CAGR

$$d = T - t_0 \ \text{(calendar days)}, \qquad CAGR = \left[\left(\frac{K(T)}{K(t_0)}\right)^{365/d} - 1\right] \times 100$$

t_0 is the first NAV date and T the latest NAV date, so d = T − t_{first}. `fund_start_date` is not used. Shown only when d ≥ 365; younger funds show the since-launch return instead. A 3Y or 5Y CAGR uses K(b) and d = T − b.

Equation 10. Fund age

$$\text{age} = \frac{\text{today} - t_{first}}{365} \ \text{years}$$

t_{first} is the date of the first NAV the fund declared (its earliest NAV row), and today is the current date. The same rule applies to every fund. Age starts on the same first NAV date as Equations 8 and 9, but runs to today instead of the latest NAV date T.

Equation 11. Calculator projection

$$r = \frac{CAGR}{100}, \quad r_m = (1+r)^{1/12} - 1, \quad FV_{lump} = A\,(1+r)^{y}, \quad FV_{SIP} = P_m\,\frac{(1+r_m)^{n} - 1}{r_m}\,(1+r_m), \quad n = 12y$$

A is the lump-sum amount, P_m the monthly SIP amount and y the number of years. A fund without a CAGR shows no projection.

5. Pseudocode

`rebuildSeries(fund)` runs after every NAV scrape and every dividend change. The return functions only read the stored `adjusted_nav`.

5.1 Choose the case and build K(t)

```
function rebuildSeries(fund):
    rows = NAV rows of fund, ascending, not is_glitch                 # from the first NAV date
    t0   = rows[0].as_of_date                                         # stored as series_start_date
    tc   = first row where composite_nav != nav                       # null = no composite

    dividends = dividends of fund with dividend_percentage > 0
    for each div: div.dividend_per_unit = div.dividend_percentage / 100 * faceValue(fund)   # Eq. 2
    for each div with a declaration_date and no admin/confirmed date:
        div.effective_date = dateFromDeclaration(fund, div)           # Eq. 3
        div.date_source = DECLARED
    for each div without a declaration_date and no admin/confirmed date:
        if tc is null or fyEnd(div) + 9 months < tc:                  # paid before any composite
            (div.effective_date, div.match_score) = bruteForceDate(fund, div)   # Eq. 4
            div.date_source = SEARCH
    dated  = dividends with an effective_date
    before = dated with effective_date < tc - 31 days

    if tc is not null and before is empty:                            # Case 1
        K = [r.composite_nav for r in rows]                           # Eq. 1, used as published
        fund.series_basis = COMPOSITE
    else if tc is not null:                                           # Hybrid
        j = index of tc in rows
        K = calculatedSeries(rows[0..j], before, unit adjustments)    # Eq. 5 up to tc
        for k = j+1 .. last: K[k] = K[k-1] * rows[k].composite_nav / rows[k-1].composite_nav   # Eq. 1a
        fund.series_basis = HYBRID
    else:
        K = calculatedSeries(rows, dated, unit adjustments)           # Eq. 5
        fund.series_basis = DIVIDENDS if dated is not empty else NAV  # Case 4 = NAV

    save K(t) into rows.adjusted_nav; rows before t0 store their own price,
    glitch rows store K(t-) * price(t) / price(t-); neither is read as the series
    fund.inception_return = (K(T) / K(t0) - 1) * 100                  # Eq. 8
    fund.cagr = Eq. 9 if days(T - t0) >= 365 else null

function calculatedSeries(rows, dividends, adjustments):              # Eq. 5
    D_on = map effective_date -> sum of dividend_per_unit
    s_on = map effective_date -> factor
    K = [rows[0].nav]
    for k = 1 .. last:
        t = rows[k].as_of_date
        K[k] = K[k-1] * (s_on.get(t, 1) * rows[k].nav + D_on.get(t, 0)) / rows[k-1].nav
    return K
```

5.2 Find the effective date

```
function dateFromDeclaration(fund, div):                              # Eq. 3
    best = argmin over NAV dates t in [div.declaration_date, +14 days] of abs(drop(t) - D)
    if best exists and drop(best) >= 0.5 * D: return best
    return first NAV date after div.declaration_date                  # flagged for admin

function bruteForceDate(fund, div):                                   # Eq. 4
    D      = div.dividend_per_unit
    fyEnd  = 30 June of the last year in div.financial_year
    dated  = fund's dividends with a declared or confirmed effective_date
    if dated is empty: dated = same AMC's dated dividends
```

```
    if dated not empty:
        o = median(days(d.effective_date - fyEnd(d)) for d in dated)
        E = fyEnd + o;  window = [E - 45 days, E + 45 days];  w = 0.5
    else:
        E = fyEnd;      window = [fyEnd - 6 months, fyEnd + 9 months];  w = 0

    best = null
    for each NAV date t in window, ascending:
        if t is a glitch row or t is used by another dividend of fund: continue
        drop = N(t-) - N(t)
        if drop < 0.5 * D: continue
        score = abs(drop - D) / D + w * abs(days(t - E)) / 45
        if best is null or score < best.score: best = (t, score)
    return best          # null -> dividend listed for admin, left out of K until dated
```

5.3 Read the returns

```
function rangeReturn(fund, range):                                    # Eq. 6
    T = latest row; cutoff = T.date minus range (calendar)
    b = latest row with date <= cutoff, else the row at t0 ("Since launch")
    points = [(r.adjusted_nav / b.adjusted_nav - 1) * 100 for r from b to T]
    return headline = points[last], points

function dayChange(fund):    return (K(T) / K(T-) - 1) * 100          # Eq. 7
function sinceLaunch(fund):  return (K(T) / K(t0) - 1) * 100          # Eq. 8

function cagr(fund, from = t0):                                       # Eq. 9
    d = days(T - from)
    if d < 365: return null
    return ((K(T) / K(from)) ^ (365 / d) - 1) * 100

function age(fund):            return days(today - first NAV date of fund) / 365   # Eq. 10

function project(fund, amount, years, mode):                          # Eq. 11
    r = cagr(fund) / 100;  if r is null: return null
    if mode == LUMP: return amount * (1 + r) ^ years
    rm = (1 + r) ^ (1/12) - 1;  n = 12 * years
    return amount * ((1 + rm) ^ n - 1) / rm * (1 + rm)
```

6. Worked examples from the database

All figures come from running the pseudocode read-only against `fintra_mf`, with NAVs up to 25 Oct 2026 (IDLC up to 19 Oct 2026).

6.1 Case 1, composite published: Ekush First Unit Fund (EFUF)

Ekush publishes a composite NAV for EFUF, so K(t) = P(t) as published.

Item	Value
t_0 (first NAV date)	25 Nov 2019, P = 10.0000
T	25 Oct 2026, P = 23.0310
$d = T - t_0$	2,526 days (6.92 years)
Since-launch return (Eq. 8)	(23.0310 / 10.0000 − 1) × 100 = **+130.31%**
CAGR (Eq. 9)	$2.30310^{365/2526} - 1$ = **12.81%**
1Y return (Eq. 6, base 25 Oct 2025)	**+17.25%**

EFUF's first NAV is 25 Nov 2019; it stays at exactly Tk 10.00 until 18 Apr 2020 and first moves on 4 Jun 2020 (Tk 10.02). Those flat rows count, so the CAGR is spread over 6.92 years. Measured from `fund_start_date` (18 Mar 2020) it would be 13.46%.

6.2 Case 1, composite published: Shanta First Income Unit Fund (SFIUF)

Shanta publishes a composite NAV for SFIUF, so K(t) = P(t) as published. t_0 = 8 Mar 2018 (P = 10.0000), T = 25 Oct 2026 (P = 17.3200), d = 3,153 days:

Measure	Value
Since-launch return (Eq. 8)	+73.20%
CAGR (Eq. 9)	6.57%
1Y return (Eq. 6, base 23 Oct 2025)	+8.73%

Shanta's composite is NAV plus total dividends paid, without reinvesting them. So over the last year it rose 8.73% while the NAV itself rose 12.61% with no dividend paid.

6.3 Case 2, calculated from declared dividends: SFIUF without its composite

Shanta records real declaration dates, so SFIUF can also be built without its composite. Equation 3 found every effective date inside the 14-day window:

FY	p	D (Tk)	t_{decl}	t_{eff}	NAV before → on t_{eff}
2023	6.00%	0.60	6 Jul 2023	6 Jul 2023	13.40 → 12.76
2022	10.00%	1.00	7 Jul 2022	7 Jul 2022	14.18 → 13.06
2021	20.50%	2.05	1 Jul 2021	15 Jul 2021	15.72 → 14.12
2019	8.00%	0.80	4 Jul 2019	4 Jul 2019	11.43 → 10.58
2018	5.00%	0.50	5 Jul 2018	19 Jul 2018	11.06 → 10.62

One step of Equation 5, on the 2021 ex-dividend day:

$$K(15\ \text{Jul}\ 2021) = 17.7047 \times \frac{14.12 + 2.05}{15.72} = 18.2115$$

Raw NAV records −10.2% that day; K records +2.9%. The calculated series ends at **+79.76%** since launch, CAGR **7.03%** and 1Y **+12.61%**, higher than the published composite's +73.20% because the calculation reinvests each dividend.

6.4 Case 3, dates found by brute force: IDLC Income Fund (IDLCIF)

IDLCIF has no composite, and all four dividend rows carry the 30 June placeholder. No IDLC fund has a dated dividend yet, so Equation 4 runs with the wide window and no timing term:

FY	p	D (Tk)	Search result	NAV before → on	Score	Admin action
2025	7.00%	0.70	17 Jul 2025	11.58 → 10.90	0.03	Confirm
2024	7.00%	0.70	18 Jul 2024	11.21 → 10.53	0.03	Confirm
2023	6.90%	0.69	No candidate	–	–	Enter interim and final separately
2022	4.85%	0.485	12 Jan 2023	drop 0.38	0.22	Reject: next year's interim

The 2022 and 2023 rows fail because the export stores yearly totals while IDLC paid a January interim and a July final (3.0% + 1.85% and 3.9% + 3.0%). With the two confirmed dates, t_0 = 24 Jun 2021 and d = 1,943 days:

Measure	NAV only (today)	With 2 confirmed dividends
Since-launch return	+17.30%	+33.13%
CAGR	3.04%	5.52%
1Y return	+4.83%	+4.83% (no dividend in the window)

6.5 Case 4, no composite and no dividend: UCB Income Plus Fund (UCBIPF)

K(t) = N(t). t_0 = 13 Jul 2023 (NAV 10.02), T = 25 Oct 2026 (NAV 15.0852), d = 1,200 days.

- Since-launch return: (15.0852 / 10.02 − 1) × 100 = **+50.55%**
- CAGR: $(15.0852 / 10.02)^{365/1200} - 1$ = **13.25%**
- Age: 1,200 / 365 = **3.29 years**

6.6 Unit restructuring under Case 1: VIPB NLI 1st Unit Fund

The NAV falls from Tk 14.95 (6 Feb 2022) to Tk 9.42 (8 Sep 2022) in a unit restructuring. VIPB's published composite already absorbs it: P/N steps from 2.1706 to 3.2450 while P moves from 32.4500 to 30.5679 (−5.8% over seven months). Case 1 therefore needs no factor s; `mf_unit_adjustments` is only for Cases 2 to 4.

7. Brute-force validation

With the payout-timing term, the search found the exact effective date for **all 14** Shanta dividends; without it, 12 of 14. Shanta is the test set because it is the only AMC with real declaration dates, so Equation 3 gives the true effective date to compare against. Each dividend was searched with its own date hidden and the timing learned from the fund's other dividends.

Search	Window	Exact matches	Misses
Season-aware (Equation 4)	E ± 45 days, timing term 0.5	14 of 14	None
Wide window, no timing term	FY_{end} − 6 months … FY_{end} + 9 months	12 of 14	Shanta Amanah FY2019 (Tk 0.13): found 31 Oct 2019, true 2 Jan 2020. Shanta Fixed Income FY2022 (Tk 0.20): found 19 May 2022, true 5 Jan 2023

Both misses are dividends of Tk 0.20 or less, the size of an ordinary day's NAV move; each matched an unrelated drop of the same size months away. The timing term removes them. The wide window remains the fallback when neither the fund nor its AMC has a dated dividend, which is why every search result needs an admin's confirmation.

The AMCs' composite NAVs were not used as a test set: their dividend steps disagree with the NAV drop by weeks in places. For VIPB NLI 1st Unit the search finds a Tk 1.42 NAV drop for a Tk 1.40 dividend on 24 Aug 2016, while the composite steps on 21 Jul 2016.

8. Data to fix before rollout

Missing dividends in Case 4 funds are the largest error left; placeholder dates matter only for funds without a composite. In order of impact on returns:

#	Problem	Funds affected	Effect on returns	Fix
1	Dividends paid but not recorded	9 Case 4 funds: CWT (4), IDLC AM Shariah, Balanced, Growth, UCB AML First, UCB Taqwa Growth	Understated: K(t) falls to raw NAV on every payout	Enter dividends from AMC memos and compliance reports; the fund moves to Case 2 or 3
2	Yearly totals stored instead of interim + final	IDLC Income	The search cannot match a split payout	Enter each payment as its own row
3	Placeholder 30 June declaration dates (`SchemaDefaults::declarationDate`)	80 rows; only IDLC Income's 4 rows feed a built series, the other 76 belong to Case 1 funds	Case 3 search instead of Case 2	Make `declaration_date` nullable, stop the default, enter real dates where known
4	Suspect rows dated 15 Sep 2025 and 29 Sep 2026, with no NAV drop after them	12 rows: CWT Sadharan Bima (2, Case 2 today), EDGE 4 funds and Ekush Stable Return (10, Case 1)	CWT Sadharan Bima's K(t) would credit dividends never paid	Verify with the AMCs, then delete

#	Problem	Funds affected	Effect on returns	Fix
5	Flat Tk 10 pre-launch rows inside the launch date	EFUF, EGF, ESRF, EDGE AI-Amin	They are the start of the series: CAGR lower, age longer	Delete the rows if the AMC did not publish them as NAVs
6	One-off NAV glitches that jump and revert	ESRF 29 Dec 2024 (+32%), EGF 31 Mar 2022 (+35%), CWT EBFG 10 Apr 2026 (−21%)	Wrong when a range starts or ends there	Flag the rows; skip them in the search and as range boundaries

Case 1 uses each published composite as it is. Shanta Amanah and Shanta First Income publish NAV plus total dividends without reinvesting them, so their returns over a year or less can read below the NAV's own change (SFIUF 1Y: +8.73% from its composite, NAV +12.61%).

9. What changes in the project

One return service builds K(t) with Equations 1 to 5 and serves Equations 6 to 11 to every page.

Where	Today	After
Fund page chart and CAGR (`funds/show.blade.php`)	`investor_return` ratio when present, else raw NAV; 365.25-day year; first NAV row as start	Equations 6 and 9 on `adjusted_nav`; 365-day year; start at t_0
Compare page chart	Raw NAV; 30/182/365-day ranges; no 5Y	Equation 6 on `adjusted_nav`; calendar ranges, same set as the fund page
Fund tiles "Since start"	`inception_return` from scrapers (a total for most AMCs, a CAGR for IDLC, empty for CWT, IDLC, UCB)	Equation 8 for every fund
Calculators (fund page and `/mutual-funds/calculators`)	`inception_return` used as a yearly rate	Equation 11 with the CAGR
Fund age (`Fund::ageInYears`)	First NAV row to latest NAV, 365.25-day year	Equation 10: first NAV date to today, 365-day year
Day change	Raw NAV change; a dividend day shows as a loss	Equation 7
Scrapers	Write `investor_return` and `inception_return`	Write NAV and composite only; the service rebuilds K(t) after each scrape
Dividend storage	30 June placeholder when the date is unknown	NULL date; effective date found by Equation 3 or 4
Admin portal	No dividend dating	Dividend entry, a review list of search results with score and NAV drop, unit adjustments, glitch flags