

# Numerology Pro — 20+ Page Report Generator (Unicode + Multilingual + Charts + Narratives)

#

-----  
-----

# Desktop app for macOS/Windows/Linux. Builds a detailed, content-rich numerology

# PDF with modules: Core numbers, Inclusion & Traits, Lo-Shu, Karmic Debts/Lessons,

# Pinnacles & Challenges, Personal Year/Month/Day, 12-month planner, Dasha-style timeline,

# Mobile analysis, Name corrections, Remedies, Appendices — reliably 20+ pages.

#

# ► One-time install:

# python3 -m pip install fpdf2 babel matplotlib pillow

#

# ► Fonts (for Unicode):

# • Latin: DejaVuSans (macOS: /System/Library/Fonts/Supplemental/DejaVuSans.ttf)

# • Hindi: place NotoSansDevanagari-Regular.ttf next to this file

#

# ► Run:

# cd ~/Desktop

# python3 numerology.py

import os

import re

import csv

import math

import json

import calendar

import tempfile

import textwrap

from dataclasses import dataclass

from datetime import date, datetime, timedelta

import tkinter as tk

from tkinter import ttk, messagebox, filedialog

# Optional pretty date (falls back if not installed)

try:

from babel.dates import format\_date as \_format\_date

def fmt\_date(d):

return \_format\_date(d, format='long', locale='en\_IN')

except Exception:

def fmt\_date(d):

return d.isoformat()

```

# PDF
try:
    from fpdf import FPDF
except Exception:
    FPDF = None

# Charts
try:
    import matplotlib
    matplotlib.use("Agg") # headless
    import matplotlib.pyplot as plt
except Exception:
    plt = None

APP_TITLE = "Numerology Pro — Report Generator"
CSV_PATH = os.path.expanduser("~/Desktop/numerology_clients.csv")
DEFAULT_SAVE_DIR = os.path.expanduser("~/Desktop")

# ----- Fonts (Unicode) ----- #
DEJAVU_FONT_CANDIDATES = [
    "/System/Library/Fonts/Supplemental/DejaVuSans.ttf",
    "/usr/share/fonts/truetype/dejavu/DejaVuSans.ttf",
    "DejaVuSans.ttf",
]
DEVANAGARI_FONT_CANDIDATES = [
    "NotoSansDevanagari-Regular.ttf",
    "/Library/Fonts/NotoSansDevanagari-Regular.ttf",
]

# ----- Numerology Maps ----- #
PYT = {
    'A':1,'B':2,'C':3,'D':4,'E':5,'F':6,'G':7,'H':8,'I':9,
    'J':1,'K':2,'L':3,'M':4,'N':5,'O':6,'P':7,'Q':8,'R':9,
    'S':1,'T':2,'U':3,'V':4,'W':5,'X':6,'Y':7,'Z':8
}

# Chaldean mapping
CHAL = {
    'A':1,'I':1,'J':1,'Q':1,'Y':1,
    'B':2,'K':2,'R':2,
    'C':3,'G':3,'L':3,'S':3,
    'D':4,'M':4,'T':4,
    'E':5,'H':5,'N':5,'X':5,
    'U':6,'V':6,'W':6,
    'O':7,'Z':7,
    'F':8,'P':8
}

```

```

VOWELS = set("AEIOU")
MASTERS = {11,22,33}
KARMIC_COMPOUNDS = {13,14,16,19}
HARMONY = {
    1:{1,3,5}, 2:{2,4,8}, 3:{1,3,5,9}, 4:{2,4,8}, 5:{1,3,5,7},
    6:{2,3,6,9}, 7:{5,7}, 8:{2,4,8}, 9:{3,6,9}
}
AUSPICIOUS_CHAL = {23,24,32,33,41,42,45,46,50,51}

# ----- Helpers ----- #
def letters_only(s: str) -> str:
    return ''.join(ch for ch in s.upper() if ch.isalpha())

def reduce_num(n: int, keep_masters=True) -> int:
    if n <= 9:
        return n
    while True:
        if keep_masters and n in MASTERS:
            return n
        n = sum(int(d) for d in str(n))
        if n <= 9 or (keep_masters and n in MASTERS):
            return n

def digits_sum_from_dob(dob: str) -> int:
    return sum(int(ch) for ch in dob if ch.isdigit())

# ----- Core Numbers ----- #
def life_path(dob: str):
    comp = digits_sum_from_dob(dob)
    return comp, reduce_num(comp, True)

def pyth_total(name: str):
    s = letters_only(name)
    comp = sum(PYT.get(ch,0) for ch in s)
    return comp, reduce_num(comp, True)

def pyth_vowels(name: str):
    s = letters_only(name)
    comp = sum(PYT.get(ch,0) for ch in s if ch in VOWELS)
    return comp, reduce_num(comp, True)

def pyth_consonants(name: str):
    s = letters_only(name)
    comp = sum(PYT.get(ch,0) for ch in s if ch not in VOWELS)
    return comp, reduce_num(comp, True)

def birthday_root(dob: str):

```

```

    day = int(dob[:2])
    return day, reduce_num(day, True)

def maturity_root(lp_root: int, expr_root: int):
    comp = lp_root + expr_root
    return comp, reduce_num(comp, True)

def chaldean_total(name: str):
    s = letters_only(name)
    comp = sum(CHAL.get(ch,0) for ch in s)
    return comp, reduce_num(comp, True)

# ----- Inclusion & Traits ----- #
def inclusion_counts(name: str):
    counts = {i:0 for i in range(1,10)}
    for ch in letters_only(name):
        n = PYT.get(ch)
        if n: counts[n]+=1
    return counts

def hidden_passions(counts):
    if not counts: return []
    mx = max(counts.values())
    return [d for d,c in counts.items() if c==mx and c>0]

def karmic_lessons(counts):
    return [d for d,c in counts.items() if c==0]

def balance_number(full_name: str):
    parts = [p for p in full_name.strip().upper().split() if p]
    total = sum(PYT.get(p[0],0) for p in parts)
    return reduce_num(total, False)

def karmic_debts_from_all(dob: str, name: str):
    res = {}
    for label, comp in (
        ("Life Path", life_path(dob)[0]),
        ("Expression", pyth_total(name)[0]),
        ("Soul Urge", pyth_vowels(name)[0]),
        ("Personality", pyth_consonants(name)[0]),
        ("Birthday", birthday_root(dob)[0]),
    ):
        if comp in KARMIC_COMPOUNDS:
            res[label]=comp
    return res

# ----- Time Cycles ----- #

```

```

def personal_year(dob: str, when: date):
    bm = int(dob[3:5]); bd = int(dob[:2])
    return reduce_num(bm + bd + when.year, True)

def personal_month(py: int, month: int):
    return reduce_num(py + month, False)

def personal_day(pm: int, day: int):
    return reduce_num(pm + day, False)

def calendar_personal_days(dob: str, when: date):
    py = personal_year(dob, when)
    pm = personal_month(py, when.month)
    days = calendar.monthrange(when.year, when.month)[1]
    return [(date(when.year, when.month, d), personal_day(pm, d)) for d in
range(1, days+1)]

```

```

# ----- Pinnacles & Challenges ----- #
def pinnacles_and_ages(dob: str):
    d = int(dob[:2]); m = int(dob[3:5]); y = int(dob[6:])
    p1 = reduce_num(d + m, True)
    p2 = reduce_num(d + y, True)
    p3 = reduce_num(p1 + p2, True)
    p4 = reduce_num(m + y, True)
    first_end = 36 - reduce_num(life_path(dob)[1], False)
    if first_end < 27: first_end = 27
    ages = [
        (0, first_end),
        (first_end+1, first_end+9),
        (first_end+10, first_end+18),
        (first_end+19, 120)
    ]
    return (p1,p2,p3,p4), ages

```

```

def challenges(dob: str):
    d = reduce_num(int(dob[:2]), False)
    m = reduce_num(int(dob[3:5]), False)
    y = reduce_num(int(dob[6:]), False)
    c1 = abs(m-d)
    c2 = abs(d-y)
    c3 = abs(c1-c2)
    c4 = abs(m-y)
    c1%=9; c2%=9; c3%=9; c4%=9
    return (c1, c2, c3, c4)

```

```

# ----- Lo-Shu (DOB) ----- #
def loshu_counts(dob: str):

```

```

counts = {i:0 for i in range(1,10)}
for ch in dob:
    if ch.isdigit():
        d = int(ch)
        if d!=0 and d in counts:
            counts[d]+=1
return counts

# ----- Mobile Number ----- #
def mobile_vibration(mobile: str):
    digits = re.sub("[^0-9]", "", mobile)
    if not digits:
        return None, None
    total = sum(int(ch) for ch in digits)
    return total, reduce_num(total, True)

# ----- Name Suggestions ----- #
CONSONANTS_FOR_H = set(list("KPRSTTDGBCM"))

def generate_name_candidates(full_name: str, max_count=24):
    base = full_name.strip()
    parts = [p for p in base.split() if p]
    if not parts: return []
    first = parts[0]; last = parts[-1] if len(parts)>1 else ""
    cands = set()
    for i in range(len(first)):
        ch = first[i].upper()
        if ch in CONSONANTS_FOR_H and (i+1==len(first) or first[i+1].lower()!
='h'):
            cands.add(' '.join([first[:i+1]+'h'+first[i+1:],*parts[1:])))
    if first and first[-1].isalpha():
        cands.add(' '.join([first+first[-1],*parts[1:])))
    swaps = {('i','y'),('y','i'),('e','i'),('i','e'),('a','aa')}
    for a,b in swaps:
        if a in first:
            cands.add(' '.join([first.replace(a,b),*parts[1:])))
    if last:
        initial = last[0].upper()
        if len(parts)==2:
            cands.add(f"{first} {initial}. {last}")
        for i,ch in enumerate(last):
            if ch.upper() in CONSONANTS_FOR_H and (i+1==len(last) or
last[i+1].lower()!='h'):
                cands.add(' '.join([first,*parts[1:-1],[last[:i+1]+'h'+last[i+1:]])]))
            break
    cands = [c for c in cands if c.lower() != base.lower()]
    return sorted(cands)[:max_count]

```

```

def score_candidate(cand: str, life_path_root: int, base_inclusion: dict):
    p_comp, p_root = pyth_total(cand)
    c_comp, c_root = chaldean_total(cand)
    harm = 0
    if life_path_root in HARMONY:
        if p_root in HARMONY[life_path_root]: harm+=6
        if c_root in HARMONY[life_path_root]: harm+=6
    penalty = -6 if p_comp in KARMIC_COMPOUNDS else 0
    bonus_chal = 4 if c_comp in AUSPICIOUS_CHAL else 0
    inc_new = inclusion_counts(cand)
    missing_base = {d for d,c in base_inclusion.items() if c==0}
    missing_new = {d for d,c in inc_new.items() if c==0}
    covered = len(missing_base - missing_new)
    inc_bonus = covered*2
    overdrive = sum(1 for v in inc_new.values() if v>=6)
    balance_penalty = -overdrive*2
    score = 50 + harm + penalty + bonus_chal + inc_bonus + balance_penalty
    return {
        'name': cand,
        'pyt_compound': p_comp,
        'pyt_root': p_root,
        'chal_compound': c_comp,
        'chal_root': c_root,
        'score': score,
        'covered_missing': list(missing_base - missing_new)
    }

```

# ----- Content Library (EN + HI narrative banks) ----- #

```

CONTENT_EN = {
    "intro": """"This premium numerology report blends Pythagorean and
Chaldean systems, the Lo-Shu grid, karmic patterns, time cycles, and practical
remedies. Treat it as a practical guide: align strengths, support weak spots, and
time key moves.""",

```

# --- Core numbers (short) ---

```

"life_path": {
    1: "Initiation, leadership, originality; act without dominating.",
    2: "Partnership, diplomacy, sensitivity; patience and cooperation.",
    3: "Creativity, joy, communication; avoid scattering energy.",
    4: "Foundations, discipline, systems; build sustainably, avoid rigidity.",
    5: "Change, freedom, versatility; embrace growth with moderation.",
    6: "Responsibility, harmony, service; care for self as you care for others.",
    7: "Analysis, wisdom, spirituality; trust intuition and logic.",
    8: "Power, achievement, stewardship; lead ethically.",
    9: "Compassion, completion, vision; serve a larger cause.",
    11: "Master 11: inspiration; ground your sensitivity.",

```

```
22: "Master 22: master builder; manifest big visions humbly.",
33: "Master 33: compassionate teacher; channel service wisely."
},
```

```
"expression": {
  1: "Executive drive and innovation.",
  2: "Diplomatic collaborator.",
  3: "Artistic communicator.",
  4: "Systematic builder.",
  5: "Agile generalist.",
  6: "Harmonizer and caretaker.",
  7: "Researcher and mystic.",
  8: "Organizer of resources.",
  9: "Humanitarian and finisher.",
  11: "Visionary inspirer.",
  22: "Master implementer.",
  33: "Healer-teacher."
},
```

```
"soul": {
  1: "Inner need to initiate and lead.",
  2: "Inner need to support and harmonize.",
  3: "Inner need to express and create.",
  4: "Inner need for order and security.",
  5: "Inner need for freedom and variety.",
  6: "Inner need to nurture and beautify.",
  7: "Inner need to understand and reflect.",
  8: "Inner need to achieve and manage.",
  9: "Inner need to serve and complete."
},
```

```
"personality": {
  1: "Comes across bold, direct, goal-oriented.",
  2: "Gentle, tactful, cooperative.",
  3: "Cheerful, expressive, social.",
  4: "Practical, reliable, methodical.",
  5: "Adaptable, adventurous, persuasive.",
  6: "Warm, responsible, aesthetic.",
  7: "Reserved, thoughtful, insightful.",
  8: "Authoritative, executive presence.",
  9: "Compassionate, broad-minded, refined."
},
```

```
# --- Core numbers (extended) ---
```

```
"life_path_long": {
  1: "Life Path 1 (Leader): You are here to originate, not imitate. Your best
seasons are those where you define a direction and take decisive, clean action.
```



Watch the shadow: forcing outcomes or equating control with safety. Practice: set one bold goal, then co-create with collaborators without micromanaging. Relationships improve when you ask for input early.",

2: "Life Path 2 (Diplomat): Harmony, listening, and partnership are your superpowers. You build trust slowly and make space for others to shine. Shadow: people-pleasing that suppresses your needs. Practice: name your preferences before consensus. Money grows via thoughtful collaborations.",

3: "Life Path 3 (Creator): Communication, creativity, and joy fuel you. Shadow: scattering energy across too many ideas. Practice: commit to one narrative per quarter and finish it. Health improves when you schedule creative play.",

4: "Life Path 4 (Builder): Systems, reliability, and craft. Shadow: rigidity and over-caution. Practice: plan the first 20 percent, then ship and iterate. Wealth compounds through durable processes.",

5: "Life Path 5 (Explorer): Change, travel, sales, learning. Shadow: impulsiveness or escapism. Practice: small, deliberate experiments with clear stop-loss; keep one anchor habit daily.",

6: "Life Path 6 (Harmonizer): Responsibility, care, and aesthetics. Shadow: control in the name of care. Practice: help by agreement, not assumption; beautify your space; accept some imperfection.",

7: "Life Path 7 (Seeker): Research, depth, and inner mastery. Shadow: isolation and overthinking. Practice: publish your learning simply; alternate solitude with grounded action.",

8: "Life Path 8 (Executive): Power, management, stewardship. Shadow: control or cynicism. Practice: lead ethically, measure what matters, share upside; consistency wins.",

9: "Life Path 9 (Humanitarian): Closure, compassion, and big vision. Shadow: giving without boundaries. Practice: complete the old before starting the new; link work to a cause; monthly release ritual.",

11: "Master 11 (Visionary): High sensitivity and inspiration. Practice: daily grounding (sleep, breath, sunlight, food) before output; translate subtle insight into helpful form.",

22: "Master 22 (Master Builder): You manifest at scale. Practice: clarify a worthy blueprint; build via teams and systems. Humility plus discipline creates realized vision.",

33: "Master 33 (Teacher-Healer): Compassion expressed practically. Practice: rest and boundaries so service stays joyful."

},

"expression\_long": "Expression shows HOW you execute your mission — your natural toolkit. When it complements your Life Path, progress feels smooth; when it conflicts, awareness and practice bridge the gap.",

# --- Inclusion guidance ---

"inclusion\_missing": {

1: "Develop personal initiative daily (micro-decisions, solo tasks).",

2: "Practice diplomacy and active listening; nurture partnership skills.",

```
3: "Schedule creative expression; writing or speaking or journaling.",
4: "Create routines and planners; build small systems.",
5: "Embrace safe change; travel and learning; avoid restlessness.",
6: "Balance duty with self-care; refine aesthetics and home.",
7: "Study and meditate; cultivate inner life and trust.",
8: "Learn money and management; lead fairly.",
9: "Volunteer or complete lingering tasks; practice letting go."
},
```

```
"inclusion_overload": {
  1: "Avoid dominance; share credit and invite input.",
  2: "Set boundaries; avoid people-pleasing loops.",
  3: "Channel talk into structured projects; avoid gossip or scatter.",
  4: "Avoid rigidity; allow creative play.",
  5: "Focus attention; limit impulsive switches.",
  6: "Care without control; avoid perfectionism expectations.",
  7: "Balance solitude with real-world action.",
  8: "Lead ethically; avoid over-control or harshness.",
  9: "Finish what you start; avoid martyr patterns."
},
```

# --- Lo-Shu helper text ---

```
"loshu_planes": {
  "mental": "Top row: strategy, logic, learning style.",
  "emotional": "Middle row: relationships, feelings, empathy.",
  "practical": "Bottom row: implementation, stamina, results."
},
```

# --- Karmic debt (short & long) ---

```
"karmic_debt_text": {
  13: "13/4 — Work-ethic, patience, process; avoid shortcuts.",
  14: "14/5 — Freedom with discipline; avoid excess or sensory traps.",
  16: "16/7 — Ego humbling, inner awakening; cultivate humility.",
  19: "19/1 — Independent integrity; lead without pride."
},
```

```
"karmic_debt_long": {
```

```
  13: ""Karmic Debt 13 — Work Ethic and Discipline
```

Meaning: In past patterns, you may have resisted hard work or sought shortcuts. This lifetime teaches persistence, patience, and humility.

Effects: Delays unless you stay consistent; lessons around responsibility.

Remedies: Daily discipline; Saturn mantra on Saturdays; navy or grey tones; donate black sesame or oil; structured exercise or yoga.""

```
  14: ""Karmic Debt 14 — Freedom and Moderation
```

Meaning: Freedom may have been misused through indulgence or lack of discipline. The lesson is balance.

Effects: Ups and downs; struggles with addictions or restlessness; instability.  
Remedies: Practice moderation; Mercury mantra on Wednesdays; green tones; donate green vegetables or stationery; periodic digital detox."",

#### 16: ""Karmic Debt 16 — Ego and Transformation

Meaning: Lessons of humility and surrender. Pride may have caused harm in past patterns.

Effects: Sudden losses or ego-crushing events; relationship tests; spiritual solitude.

Remedies: Chant Om Namah Shivaya; consider Cat's Eye after testing; meditation and journaling; time near water; donate blankets or spiritual books."",

#### 19: ""Karmic Debt 19 — Independence and Integrity

Meaning: Independence may have been misused through selfishness or misuse of authority. Lead with fairness and humility now.

Effects: Struggles with authority; isolation if prideful; tests of honesty and responsibility.

Remedies: Serve while leading; Sun mantra on Sundays; ruby or gold accents; donate wheat or copper; acts of service without expectation.""

},

# --- Year / month / day meanings ---

"year\_meanings": {

- 1: "Year 1 — Begin boldly, plant seeds, define direction.",
- 2: "Year 2 — Partner, refine, be patient; nurture what was planted.",
- 3: "Year 3 — Create, communicate, expand social reach.",
- 4: "Year 4 — Build structures and systems; sustained work.",
- 5: "Year 5 — Change, travel, pivot; stay balanced.",
- 6: "Year 6 — Family, responsibility, beauty; repair and reconcile.",
- 7: "Year 7 — Study, reflect, reset; inner mastery.",
- 8: "Year 8 — Execute, manage, advance; money and leadership themes.",
- 9: "Year 9 — Completion, release, compassion; finish cycles."

},

"challenge\_meanings": {

0: "Challenge 0 — broad lesson: develop wholeness, compassion, flexibility.",

- 1: "Challenge 1 — confidence with tact; self-direction without ego.",
- 2: "Challenge 2 — patience, boundaries, cooperation.",
- 3: "Challenge 3 — truthful expression; avoid self-criticism.",
- 4: "Challenge 4 — persistence and planning vs rigidity.",
- 5: "Challenge 5 — focus and commitment vs escapism.",
- 6: "Challenge 6 — care without control; balanced duty.",
- 7: "Challenge 7 — trust and connection vs isolation.",
- 8: "Challenge 8 — ethics in power, money, and management.",
- 9: "Challenge 9 — compassion and completion without martyrdom."

},

# --- Pinnacles & Challenges (extended chapters) ---

"pinnacles\_long": {

1: "Pinnacle 1 — New Beginnings and Independence. Theme: leadership and bold moves. Opportunities: start ventures, pursue unique paths. Shadow: ego or isolation. Remedies: humility, share credit, Sun mantra, red or gold accents.",

2: "Pinnacle 2 — Cooperation and Patience. Theme: relationships and diplomacy. Opportunities: partnerships, collaborations. Shadow: over-sensitivity or dependency. Remedies: Moon rituals, water and rest, white or silver.",

3: "Pinnacle 3 — Expression and Creativity. Theme: communication and joy. Opportunities: teaching, publishing, art. Shadow: scattered energy. Remedies: Saraswati mantra, journaling, yellow attire, donate books.",

4: "Pinnacle 4 — Discipline and Foundations. Theme: hard work and planning. Opportunities: build systems, long-term investments. Shadow: rigidity or stress. Remedies: Saturn rituals, yoga routines, navy or grey, donate oil.",

5: "Pinnacle 5 — Change and Freedom. Theme: adaptability and travel. Opportunities: expansion and networking. Shadow: restlessness or addictions. Remedies: Mercury mantra, breathwork, green attire, donate stationery.",

6: "Pinnacle 6 — Family and Harmony. Theme: home, love, and responsibility. Opportunities: strong bonds and creative beauty. Shadow: control or perfectionism. Remedies: Venus rituals, art and music, white or pink, donate sweets.",

7: "Pinnacle 7 — Wisdom and Spirituality. Theme: study and awakening. Opportunities: healing, research, academia. Shadow: isolation or skepticism. Remedies: Shiva mantra, meditation, violet tones, time near water.",

8: "Pinnacle 8 — Power and Achievement. Theme: authority, wealth, mastery. Opportunities: leadership roles, financial success. Shadow: greed or harshness. Remedies: Saturn mantra, ethical practices, dark blue or black, charity.",

9: "Pinnacle 9 — Completion and Service. Theme: closure and humanitarian work. Opportunities: social impact and legacy. Shadow: martyrdom or weak boundaries. Remedies: Mars rituals, forgiveness, red or maroon, charity."

},

"challenges\_long": {

0: "Challenge 0 — Universal test. Experience life fully and cultivate balance. Remedies: meditation, balanced diet, volunteering.",

1: "Challenge 1 — Ego and confidence. Build self-reliance without arrogance. Remedies: Sun mantra, leadership practice, humility.",

2: "Challenge 2 — Sensitivity and boundaries. Cooperate without dependency. Remedies: Moon mantra, assertiveness practice.",

3: "Challenge 3 — Expression and self-worth. Communicate authentically. Remedies: journaling, throat-care practices, yellow tones.",

4: "Challenge 4 — Hard work and rigidity. Learn discipline with flexibility. Remedies: Saturn mantra, routines, grounding yoga.",

5: "Challenge 5 — Freedom vs responsibility. Balance exploration with discipline. Remedies: Mercury mantra, digital detox, green clothing.",

6: "Challenge 6 — Care vs control. Love without attachment. Remedies: Venus mantra, art therapy, boundaries in family.",

7: "Challenge 7 — Introspection vs isolation. Gain wisdom without withdrawal. Remedies: Shiva mantra, study groups, water meditation.",

8: "Challenge 8 — Power and karma. Handle money and authority ethically. Remedies: Saturn mantra, donate food or oil, fair leadership.",

9: "Challenge 9 — Compassion and completion. Serve without martyrdom. Remedies: Mars mantra, forgiveness rituals, charity."

},

# --- Predictions templates by Life Path ---

"pred\_templates": {

  "career": {

    1: "Lead a lean project; pick a visible initiative. Avoid micromanagement.",

    2: "Form alliances; co-author and mediate. Grow influence via partnerships.",

    3: "Publish or perform; pitch ideas clearly; build a portfolio.",

    4: "Standardize processes; earn through reliability and operations.",

    5: "Sales or marketing or travel; ship experiments quickly.",

    6: "Design or aesthetics or service; reputation via consistent care.",

    7: "Analytics or research or education; long-form study pays off.",

    8: "Management or finance or operations; negotiate for fair leverage.",

    9: "Social impact or media or mentoring; finish legacy projects."

  },

  "relationships": {

    1: "Show initiative but ask; avoid dominance; be curious.",

    2: "Prioritize listening; co-decide; gentle honesty.",

    3: "Express feelings; schedule fun; avoid sarcasm.",

    4: "Plan quality time; be flexible inside routines.",

    5: "Novelty plus loyalty; avoid impulsive extremes.",

    6: "Care without control; aesthetics and comfort matter.",

    7: "Share inner world gradually; respect solitude.",

    8: "Fairness and affection; shared goals and budgets.",

    9: "Release the past; serve together; cultivate forgiveness."

  },

  "health": {

    1: "Channel intensity via exercise; watch blood pressure.",

    2: "Hydration and sleep; calm the nervous system.",

    3: "Throat and voice care; breathing and posture.",

    4: "Bones and teeth; steady movement; anti-rigidity.",

    5: "Nervous or digestive balance; breathwork.",

    6: "Sugar balance; music and restorative arts.",

```

7: "Rest and eyes; meditation; time in nature.",
8: "Joints and skin; strength with mobility.",
9: "Inflammation control; paced cardio."
},
"money": {
  1: "Negotiate; monetize leadership and IP.",
  2: "Referral income through partners.",
  3: "Content or product launches; small recurring revenue.",
  4: "Slow-and-steady systems produce cashflow.",
  5: "Multiple small bets; diversify.",
  6: "Reputation compounding; premium service.",
  7: "Specialist advisory or courses.",
  8: "Scale operations; ethical leverage.",
  9: "Cause-linked offerings; close overdue dues."
}
},

```

# --- Personal Month and Day narratives ---

```

"personal_month_long": {
  1: "Month 1 — Fresh starts and initiative. Launch projects, set bold goals.
Avoid impatience or ego clashes. Affirm: I courageously step into new
opportunities.",
  2: "Month 2 — Patience and partnerships. Build alliances and do detail
work. Delay big launches. Affirm: I grow with patience and harmony.",
  3: "Month 3 — Expression and creativity. Communicate and teach. Avoid
scattering energy. Affirm: I express with clarity and joy.",
  4: "Month 4 — Hard work and structure. Systems and savings matter.
Avoid rigidity. Affirm: I create stability through steady effort.",
  5: "Month 5 — Change and adventure. Network and learn. Avoid risky
impulses. Affirm: I embrace change with balance.",
  6: "Month 6 — Responsibility and relationships. Care for family and
beautify spaces. Avoid over-control. Affirm: I care with balance and grace.",
  7: "Month 7 — Reflection and inner growth. Meditate and study; do not
rush. Affirm: I trust the wisdom that comes in silence.",
  8: "Month 8 — Achievement and results. Take bold financial steps; be
ethical. Affirm: I achieve success with fairness and strength.",
  9: "Month 9 — Closure and release. Finish projects; forgive. Avoid starting
big things now. Affirm: I release the old and welcome the new."
},

```

```

"personal_day_long": {
  1: "Day 1 — Action and initiative. Best for starting tasks and decisions.
Avoid ego clashes. Affirm: I step forward with courage and wisdom.",
  2: "Day 2 — Cooperation and sensitivity. Best for teamwork and healing.
Avoid hesitation. Affirm: I create harmony with patience.",
  3: "Day 3 — Creativity and expression. Best for writing and presentations.
Avoid gossip. Affirm: I express myself with clarity and joy.",

```

4: "Day 4 — Work and foundations. Best for planning and routines. Avoid pessimism. Affirm: I build stability step by step.",  
5: "Day 5 — Change and adventure. Best for travel and networking. Avoid scattered energy. Affirm: I embrace change with balance.",  
6: "Day 6 — Love and responsibility. Best for family and beauty projects. Avoid perfectionism. Affirm: I serve with love and balance.",  
7: "Day 7 — Reflection and wisdom. Best for meditation and study. Avoid isolation. Affirm: I grow through reflection and inner wisdom.",  
8: "Day 8 — Power and results. Best for finance and career moves. Avoid harshness. Affirm: I achieve success with fairness.",  
9: "Day 9 — Completion and compassion. Best for closure and charity. Avoid clinging to the past. Affirm: I release and move forward with compassion."  
},

# --- Lifestyle guidance (diet, yoga, crystals, routine) ---

"lifestyle\_long": {

1: "Number 1 — Sun: Diet warm fresh foods (wheat, honey, oranges). Yoga: Surya Namaskar, pranayama. Crystals: Sunstone, Ruby. Routine: Morning sunlight, leadership journaling, goal-setting.",

2: "Number 2 — Moon: Diet milk, rice, cucumber, melons. Yoga: Chandra Namaskar, cooling pranayama. Crystals: Moonstone, Pearl. Routine: Evening meditation, calming music, journal emotions.",

3: "Number 3 — Jupiter: Diet ghee, yellow pulses, bananas, turmeric. Yoga: Mantra chanting, Kapalabhati. Crystals: Yellow Sapphire, Citrine. Routine: Teaching, writing, gratitude journaling.",

4: "Number 4 — Rahu: Diet root vegetables and nuts, avoid junk. Yoga: Tadasana, Vajrasana. Crystals: Hessonite, Smoky Quartz. Routine: Fixed schedule, to-do lists, digital detox.",

5: "Number 5 — Mercury: Diet green vegetables, mint, apples. Yoga: Pranayama and dynamic flows. Crystals: Emerald, Green Aventurine. Routine: Travel, networking, language practice.",

6: "Number 6 — Venus: Diet fruits and dairy; sweets in moderation. Yoga: Heart-opening flows. Crystals: Diamond, Rose Quartz. Routine: Music, art, family bonding, self-care rituals.",

7: "Number 7 — Ketu: Diet herbal teas, light vegetarian; weekly fast. Yoga: Meditation, Yoga Nidra, pranayama. Crystals: Cat's Eye, Amethyst. Routine: Solitude for reflection, spiritual study.",

8: "Number 8 — Saturn: Diet barley, sesame, black gram, greens; avoid alcohol. Yoga: Grounding and balance asanas. Crystals: Blue Sapphire, Black Tourmaline. Routine: Charity Saturdays, ethical leadership.",

9: "Number 9 — Mars: Diet red fruits, lentils, beetroot, iron-rich foods. Yoga: Dynamic yoga, Surya Namaskar, martial arts. Crystals: Red Coral, Garnet. Routine: Exercise, forgiveness practice, social service."

},

# --- Crystals + Colour therapy ---

"crystal\_colour": {

- 1: "Colour: red, gold, orange (Sundays). Crystals: Ruby, Sunstone. Energise in morning sunlight; wear on right hand.",
- 2: "Colour: white, cream, silver (Mondays). Crystals: Pearl, Moonstone. Energise in full moonlight; wear little finger (silver).",
- 3: "Colour: yellow, saffron (Thursdays). Crystals: Yellow Sapphire, Citrine. Energise in turmeric water; wear index finger (gold).",
- 4: "Colour: grey, smoky blue (Saturdays). Crystals: Hessonite, Smoky Quartz. Keep in iron bowl overnight; wear middle finger (silver).",
- 5: "Colour: green (Wednesdays). Crystals: Emerald, Green Aventurine. Energise in basil water; wear little finger (gold or silver).",
- 6: "Colour: pink, white, pastels (Fridays). Crystals: Diamond, Rose Quartz. Energise in rose water; wear on ring finger.",
- 7: "Colour: violet, purple. Crystals: Cat's Eye, Amethyst. Energise with mantra Om Ketave Namah; wear middle finger (silver).",
- 8: "Colour: dark blue, black (Saturdays). Crystals: Blue Sapphire, Black Tourmaline. Energise in sesame oil; wear middle finger (iron).",
- 9: "Colour: red, maroon (Tuesdays). Crystals: Red Coral, Garnet. Energise in pomegranate juice; wear ring finger (gold or copper)."

},

# --- Remedies master toolkit (mantras, donations, fasting, daily practices)

---

"remedies\_master": {

- 1: "Sun: Mantra Om Hram Hreem Hraum Sah Suryaya Namah (11 or 21 Sundays). Donate wheat, jaggery, copper. Light Sunday fast. Practices: sunlight, confidence affirmations.",
- 2: "Moon: Mantra Om Chandraya Namah (11 Mondays). Donate rice, milk, white clothes. Avoid salty food on Mondays. Practices: evening meditation, gratitude journaling.",
- 3: "Jupiter: Mantra Om Brim Brihaspataye Namah (19 Thursdays). Donate turmeric, yellow sweets, books. Light Thursday fast. Practices: teaching, chanting, gratitude writing.",
- 4: "Rahu: Mantra Om Ram Rahave Namah (108 Saturdays). Donate blanket, iron, mustard oil. Simple Saturday meals. Practices: grounding yoga, digital detox, discipline tasks.",
- 5: "Mercury: Mantra Om Bum Budhaya Namah (9 Wednesdays). Donate greens and stationery. Light green-food fast. Practices: journaling ideas, language learning, short travel.",
- 6: "Venus: Mantra Om Shum Shukraya Namah (16 Fridays). Donate sweets, curd, perfumes, white clothes. Light Friday fast. Practices: music, art, self-care with boundaries.",
- 7: "Ketu: Mantra Om Kem Ketave Namah (108 on Tue/Sat). Donate blankets, brown clothes, seven grains. Weekly fruit/water fast. Practices: meditation, study, nature walks.",
- 8: "Saturn: Mantra Om Sham Shanicharaya Namah (108 Saturdays). Donate black clothes, sesame, oil. Saturday fast with barley/black gram. Practices: ethical leadership, service.",



9: "Mars: Mantra Om Kram Krim Kraum Sah Bhaumaya Namah (9 or 27 Tuesdays). Donate red lentils, jaggery, red cloth. Simple satvik fast. Practices: exercise, service, forgiveness."

},

# --- Appendix: Archetypes quick reference ---

"appendix\_number": {

1: "1 — Initiative, autonomy, will. Career: entrepreneurship, leadership. Relationships: lead without dominating; praise others. Health: watch blood pressure; channel energy via exercise. Remedies: Sunday sunlight; affirm I initiate with integrity.",

2: "2 — Cooperation, rhythm, sensitivity. Career: mediation, HR, design. Relationships: empathy plus boundaries. Health: hydration and sleep. Remedies: Moon meditation Mondays; silver or white tones.",

3: "3 — Expression, art, optimism. Career: media, writing, teaching. Relationships: communicate feelings; avoid sarcasm. Health: throat care. Remedies: chanting, journaling, yellow on Thursdays.",

4: "4 — Order, practicality, systems. Career: engineering, ops, real estate. Relationships: be flexible; schedule quality time. Health: bones and teeth. Remedies: routines, earthy colors, Saturday structure.",

5: "5 — Change, travel, sales, tech. Relationships: variety within loyalty; avoid escapism. Health: nerves and digestion. Remedies: breathwork, short trips, Mercury-friendly greens.",

6: "6 — Harmony, service, beauty. Career: caregiving, hospitality, design. Relationships: care without control. Health: sugar balance. Remedies: aesthetics, music, Friday charity.",

7: "7 — Research, spirituality, depth. Career: analytics, academia, healing arts. Relationships: share inner world; avoid isolation. Health: rest and eyes. Remedies: water bodies, meditation, info fasts.",

8: "8 — Management, finance, authority. Relationships: tenderness and fairness. Health: joints and skin. Remedies: ethical leadership, Saturday charity, navy or black grounding.",

9: "9 — Compassion, global vision, finishing. Relationships: release past; serve together. Health: inflammation. Remedies: Mars discipline, red in moderation, Tuesday service."

}

}

CONTENT\_HI = {

"intro": """"यह प्रीमियम अंक ज्योतिष रिपोर्ट पायथागोरियन और काल्डियन प्रणालियों, लो-शु ग्रिड, कर्मिक पैटर्न, समय चक्र और व्यावहारिक उपायों का मिश्रण है। इसे एक मार्गदर्शक की तरह मानें: अपनी ताकतों को संरेखित करें, कमजोरियों को सहारा दें और सही समय पर सही कदम उठाएँ।""",

# --- मुख्य अंक (संक्षिप्त) ---

```
"life_path": {
  1: "आरंभ, नेतृत्व, मौलिकता; प्रभुत्व से नहीं, निष्ठा से कार्य करें।",
  2: "साझेदारी, कूटनीति, संवेदनशीलता; धैर्य और सहयोग।",
  3: "रचनात्मकता, आनंद, संवाद; ऊर्जा बिखराव से बचें।",
  4: "नींव, अनुशासन, प्रणाली; टिकाऊ ढंग से निर्माण करें।",
  5: "परिवर्तन, स्वतंत्रता, बहुमुखी प्रतिभा; संतुलित विकास अपनाएँ।",
  6: "जिम्मेदारी, सामंजस्य, सेवा; दूसरों की तरह स्वयं का भी ख्याल।",
  7: "विश्लेषण, ज्ञान, आध्यात्मिकता; तर्क और अंतर्ज्ञान पर भरोसा करें।",
  8: "शक्ति, उपलब्धि, प्रबंधन; नैतिक नेतृत्व करें।",
  9: "करुणा, पूर्णता, दृष्टि; बड़े उद्देश्य की सेवा।",
  11: "मास्टर 11: प्रेरणा; संवेदनशीलता को स्थिर करें।",
  22: "मास्टर 22: महान निर्माता; विनम्रता से बड़े विज्ञान साकार करें।",
  33: "मास्टर 33: करुणामय शिक्षक; सेवा को विवेक से दिशा दें।"
},
```

```
"expression": {
  1: "नवाचार और कार्यकारी क्षमता।",
  2: "कूटनीतिक सहयोगी।",
  3: "कलात्मक संप्रेषक।",
  4: "संगठित निर्माता।",
  5: "चुस्त और बहु-प्रतिभाशाली।",
  6: "सामंजस्यकारी और देखभाल करने वाला।",
  7: "अनुसंधानकर्ता और रहस्यवादी।",
  8: "संसाधन संयोजक।",
  9: "मानवतावादी और पूर्णकर्ता।",
  11: "दूरदर्शी प्रेरक।",
  22: "महान क्रियान्वयनकर्ता।",
  33: "चिकित्सक-शिक्षक।"
},
```

```
"soul": {
  1: "भीतर से नेतृत्व और शुरुआत की आवश्यकता।",
  2: "भीतर से सहयोग और सामंजस्य की आवश्यकता।",
  3: "भीतर से अभिव्यक्ति और रचना की आवश्यकता।",
  4: "भीतर से व्यवस्था और सुरक्षा की आवश्यकता।",
  5: "भीतर से स्वतंत्रता और विविधता की आवश्यकता।",
  6: "भीतर से देखभाल और सौंदर्य की आवश्यकता।",
  7: "भीतर से समझ और आत्मचिंतन की आवश्यकता।",
  8: "भीतर से उपलब्धि और प्रबंधन की आवश्यकता।",
  9: "भीतर से सेवा और पूर्णता की आवश्यकता।"
},
```

"personality": {

- 1: "दृढ़, सीधा, लक्ष्य-उन्मुख व्यक्तित्व।",
- 2: "कोमल, विनम्र, सहयोगी।",
- 3: "प्रसन्नचित्त, अभिव्यक्तिपूर्ण, सामाजिक।",
- 4: "व्यावहारिक, भरोसेमंद, व्यवस्थित।",
- 5: "अनुकूलनीय, साहसी, प्रभावशाली।",
- 6: "सौम्य, जिम्मेदार, सौंदर्यप्रिय।",
- 7: "संयमी, चिंतनशील, गहन।",
- 8: "प्रभावशाली, कार्यकारी उपस्थिति।",
- 9: "करुणामय, व्यापक दृष्टिकोण वाला, परिष्कृत।"

},

# --- मुख्य अंक (विस्तृत) ---

"life\_path\_long": {

- 1: "लाइफ पाथ 1 (नेता): आपका उद्देश्य मौलिकता है। श्रेष्ठ समय वे होते हैं जब आप दिशा तय कर निर्णायक कदम उठाते हैं। छाया पक्ष: परिणामों पर ज़बरदस्ती करना। अभ्यास: साहसिक लक्ष्य चुनें, दूसरों के साथ मिलकर बिना नियंत्रण के आगे बढ़ें।",
- 2: "लाइफ पाथ 2 (कूटनीतिज्ञ): साझेदारी और सामंजस्य आपकी शक्ति है। छाया पक्ष: अपनी ज़रूरतों को दबाना। अभ्यास: सहमति से पहले अपनी प्राथमिकताएँ बताएं।",
- 3: "लाइफ पाथ 3 (निर्माता): संवाद और रचनात्मकता आपकी ऊर्जा है। छाया पक्ष: अनेक विचारों में बिखराव। अभ्यास: एक कहानी चुनें और उसे पूरा करें।",
- 4: "लाइफ पाथ 4 (निर्माता): प्रणाली और विश्वसनीयता। छाया पक्ष: कठोरता। अभ्यास: आंशिक योजना बनाएं और कार्य प्रारंभ करें।",
- 5: "लाइफ पाथ 5 (अन्वेषक): परिवर्तन और यात्रा। छाया पक्ष: आवेग या पलायन। अभ्यास: छोटे, नियंत्रित प्रयोग करें।",
- 6: "लाइफ पाथ 6 (सामंजस्यकारी): जिम्मेदारी और सौंदर्य। छाया पक्ष: देखभाल के नाम पर नियंत्रण। अभ्यास: सौंदर्य का आनंद लें और अपूर्णता स्वीकारें।",
- 7: "लाइफ पाथ 7 (अन्वेषक): शोध और आध्यात्म। छाया पक्ष: एकांत और अति-चिंतन। अभ्यास: ज्ञान प्रकाशित करें।",
- 8: "लाइफ पाथ 8 (प्रबंधक): शक्ति और प्रबंधन। छाया पक्ष: नियंत्रण और नकारात्मकता। अभ्यास: नैतिक नेतृत्व करें।",
- 9: "लाइफ पाथ 9 (मानवतावादी): करुणा और दृष्टि। छाया पक्ष: सीमाओं के बिना देना। अभ्यास: पुराने कार्य पूरे करें।",
- 11: "मास्टर 11 (द्रष्टा): उच्च संवेदनशीलता और प्रेरणा। अभ्यास: दैनिक आधार पर ग्राउंडिंग करें।",
- 22: "मास्टर 22 (महान निर्माता): बड़े पैमाने पर साकार करना। अभ्यास: योग्य योजना बनाएँ और टीमवर्क से कार्य करें।",
- 33: "मास्टर 33 (शिक्षक-चिकित्सक): करुणा का व्यावहारिक रूप। अभ्यास: सेवा करते समय आराम और सीमाएँ बनाए रखें।"

},

"expression\_long": "एक्सप्रेशन दर्शाता है कि आप अपना मिशन कैसे पूरा करते हैं — आपका प्राकृतिक टूलकिट। जब यह आपके लाइफ पाथ के साथ सामंजस्य रखता है, तो प्रगति आसान होती है; अन्यथा अभ्यास और जागरूकता से संतुलन बनाना पड़ता है।",

# --- इन्क्लूजन मार्गदर्शन ---

```
"inclusion_missing": {  
  1: "प्रतिदिन पहल करें; छोटे निर्णय खुद लें।",  
  2: "सुनने और सहयोग का अभ्यास करें।",  
  3: "रचनात्मक लेखन या बोलने का अभ्यास करें।",  
  4: "रूटीन और योजनाएँ बनाएँ।",  
  5: "सुरक्षित परिवर्तन अपनाएँ; यात्रा और सीखें।",  
  6: "कर्तव्य और आत्म-देखभाल में संतुलन।",  
  7: "अध्ययन और ध्यान करें।",  
  8: "धन और प्रबंधन सीखें।",  
  9: "स्वेच्छा से सेवा करें और पुराने कार्य पूरे करें।"  
},
```

```
"inclusion_overload": {  
  1: "प्रभुत्व से बचें; दूसरों को शामिल करें।",  
  2: "सीमाएँ तय करें; अत्यधिक सहमति से बचें।",  
  3: "बातचीत को परियोजना में केंद्रित करें।",  
  4: "कठोरता से बचें; रचनात्मकता अपनाएँ।",  
  5: "ध्यान केंद्रित करें; बार-बार बदलाव से बचें।",  
  6: "देखभाल में संतुलन; पूर्णतावाद से बचें।",  
  7: "एकांत और क्रिया में संतुलन।",  
  8: "नैतिक नेतृत्व करें; अति-नियंत्रण से बचें।",  
  9: "शुरू किया कार्य पूरा करें।"  
},
```

# --- लो-शु मार्गदर्शन ---

```
"loshu_planes": {  
  "mental": "ऊपरी पंक्ति: रणनीति और तर्क।",  
  "emotional": "मध्य पंक्ति: संबंध और सहानुभूति।",  
  "practical": "निचली पंक्ति: क्रियान्वयन और परिणाम।"  
},
```

# --- कर्मिक ऋण ---

```
"karmic_debt_text": {  
  13: "13/4 — धैर्य और मेहनत; शॉर्टकट से बचें।",  
  14: "14/5 — स्वतंत्रता के साथ अनुशासन।",  
  16: "16/7 — अहं का त्याग; आध्यात्मिक सबक।",  
}
```

```

19: "19/1 — ईमानदारी से नेतृत्व करें।"
},

"karmic_debt_long": {
  13: "कर्मिक ऋण 13 — अनुशासन और परिश्रम। अर्थ: धैर्य और निरंतरता सीखना। उपाय: शनिवारीय उपवास, काले तिल दान।",
  14: "कर्मिक ऋण 14 — स्वतंत्रता और संतुलन। अर्थ: अनुशासन से संतुलन पाना। उपाय: बुधवार व्रत, हरा दान।",
  16: "कर्मिक ऋण 16 — अहं और रूपांतरण। अर्थ: विनम्रता और आत्मनिरीक्षण। उपाय: ओम नमः शिवाय, ध्यान, जल निकट समय।",
  19: "कर्मिक ऋण 19 — स्वतंत्रता और ईमानदारी। अर्थ: नेतृत्व में विनम्रता। उपाय: रविवार सूर्य उपासना, गेहूँ और तांबा दान।"
},

# --- वर्ष/माह ---
"year_meanings": {
  1: "वर्ष 1 — नई शुरुआत।",
  2: "वर्ष 2 — साझेदारी और धैर्य।",
  3: "वर्ष 3 — अभिव्यक्ति और रचना।",
  4: "वर्ष 4 — अनुशासन और संरचना।",
  5: "वर्ष 5 — परिवर्तन और यात्रा।",
  6: "वर्ष 6 — परिवार और जिम्मेदारी।",
  7: "वर्ष 7 — अध्ययन और आत्मचिंतन।",
  8: "वर्ष 8 — प्रबंधन और उपलब्धि।",
  9: "वर्ष 9 — समापन और करुणा।"
},

"challenge_meanings": {
  0: "चुनौती 0 — सम्पूर्णता और संतुलन।",
  1: "चुनौती 1 — आत्मविश्वास बिना अहंकार।",
  2: "चुनौती 2 — सहयोग में सीमाएँ।",
  3: "चुनौती 3 — सच्ची अभिव्यक्ति।",
  4: "चुनौती 4 — कठोरता बनाम लचीलापन।",
  5: "चुनौती 5 — स्वतंत्रता बनाम जिम्मेदारी।",
  6: "चुनौती 6 — देखभाल बनाम नियंत्रण।",
  7: "चुनौती 7 — आत्मचिंतन बनाम अलगाव।",
  8: "चुनौती 8 — धन और शक्ति का नैतिक उपयोग।",
  9: "चुनौती 9 — करुणा और समापन बिना बलिदान।"
},
}

```

```

# ----- Narrative Engine ----- #
def _pick_lang(content_en, content_hi, lang):
    return content_hi if lang == 'hi' else content_en

def make_core_narrative(payload, lang):
    C = _pick_lang(CONTENT_EN, CONTENT_HI, lang)
    core = payload['core']
    lines = []
    lines.append(( "Life Path" if lang=='en' else "लाइफ पाथ",
                   core['life_path'][1], C["life_path"].get(core['life_path'][1], "") ))
    for label, key, lib in [
        ("Expression","expression","expression"),
        ("Soul Urge","soul","soul"),
        ("Personality","personality","personality"),
    ]:
        root = core[key][1]
        if lang=='en':
            txt = CONTENT_EN.get(lib, {}).get(root, "")
            lbl = label
        else:
            txt = CONTENT_EN.get(lib, {}).get(root, "")
            lbl = {"Expression":"एक्सप्रेसन","Soul Urge":"सोल
अर्जी","Personality":"पर्सनालिटी"}[label]
        lines.append((lbl, root, txt))
    return lines

def make_inclusion_narrative(payload, lang):
    counts = payload['inclusion']
    missing = payload['missing']
    over = [d for d,c in counts.items() if c>=6]
    parts = []
    if missing:
        for d in missing:
            tip = CONTENT_EN["inclusion_missing"].get(d, "")
            parts.append((f"Missing {d}" if lang=='en' else f"मिसिंग {d}") + ": " + tip)
    if over:
        for d in over:
            tip = CONTENT_EN["inclusion_overload"].get(d, "")
            parts.append((f"Overloaded {d}" if lang=='en' else f"ओवरलोड {d}") + ": "
+ tip)
    if not parts:
        parts.append("Balanced inclusion profile; maintain steady habits and
periodic review."
                    if lang=='en' else
                    "इन्क्लूजन संतुलित है; स्थिर आदतें बनाए रखें और समय-समय पर समीक्षा करें।")

```

```
return parts
```

```
def make_loshu_narrative(payload, lang):
    C = _pick_lang(CONTENT_EN, CONTENT_HI, lang)
    los = payload['loshu']
    mental = los.get(4,0)+los.get(9,0)+los.get(2,0)
    emotional = los.get(3,0)+los.get(5,0)+los.get(7,0)
    practical = los.get(8,0)+los.get(1,0)+los.get(6,0)
    if lang=='en':
        return [
            f"Mental plane: {mental} — {CONTENT_EN['loshu_planes']['mental']}",
            f"Emotional plane: {emotional} — {CONTENT_EN['loshu_planes']",
            f"Practical plane: {practical} — {CONTENT_EN['loshu_planes']",
        ]
    else:
        return [
            f"मानसिक तल: {mental} — रणनीति/अध्ययन शैली।",
            f"भावनात्मक तल: {emotional} — सम्बन्ध/सहानुभूति।",
            f"व्यावहारिक तल: {practical} — क्रियान्वयन/परिणाम।",
        ]
```

```
def make_karmic_debt_narrative(payload, lang):
    kd = payload['karmic_debts']
    if not kd:
        return ["No karmic debt compounds detected in key positions."
                if lang=='en' else "मुख्य स्थानों में कोई कर्मिक ऋण सम्मिश्र संख्या नहीं मिली।"]
    out = []
    for where, comp in kd.items():
        txt = CONTENT_EN["karmic_debt_text"].get(comp, "Karmic learning
        indicated; practice awareness and discipline.")
        if lang=='en':
            out.append(f"{where}: {txt}")
        else:
            out.append(f"{where}: {txt}")
    return out
```

```
def make_cycles_narrative(payload, lang):
    py = payload['py']
    if lang=='en':
        ym = CONTENT_EN["year_meanings"].get(py, "")
        return [f"Personal Year {py}: {ym}"]
    else:
        ym = CONTENT_HI["year_meanings"].get(py, "")
        return [f"पर्सनल ईयर {py}: {ym}"]
```

```

def make_predictions(payload, lang):
    lp = payload['core']['life_path'][1]
    P = CONTENT_EN["pred_templates"]
    preds = {
        "Career" if lang=='en' else "करियर": P["career"][lp],
        "Relationships" if lang=='en' else "रिलेशनशिप": P["relationships"][lp],
        "Health" if lang=='en' else "स्वास्थ्य": P["health"][lp],
        "Money" if lang=='en' else "धन": P["money"][lp],
    }
    return preds

def depth_pages(depth):
    if depth == 'concise': return 8
    if depth == 'standard': return 14
    return 22 # detailed

# ----- Charts (PNG) ----- #
def chart_inclusion(counts: dict) -> str:
    if plt is None:
        raise RuntimeError("matplotlib not installed. Run: python3 -m pip install matplotlib pillow")
    xs = list(range(1,10))
    ys = [counts.get(i,0) for i in xs]
    fig = plt.figure(figsize=(5.2,2.8), dpi=200)
    plt.bar(xs, ys)
    plt.xticks(xs)
    plt.title("Inclusion Chart")
    plt.xlabel("Digits 1-9")
    plt.ylabel("Frequency")
    tmp = tempfile.mkstemp(suffix="_incl.png")[1]
    plt.tight_layout()
    plt.savefig(tmp)
    plt.close(fig)
    return tmp

def chart_loshu(counts: dict) -> str:
    if plt is None:
        raise RuntimeError("matplotlib not installed. Run: python3 -m pip install matplotlib pillow")
    fig = plt.figure(figsize=(3.6,3.6), dpi=200)
    ax = fig.add_subplot(111)
    ax.set_xticks([0,1,2,3]); ax.set_yticks([0,1,2,3])
    ax.grid(True)
    ax.set_xlim(0,3); ax.set_ylim(0,3)
    ax.set_xticklabels([]); ax.set_yticklabels([])
    pos = {

```



```

    4:(0,0), 9:(1,0), 2:(2,0),
    3:(0,1), 5:(1,1), 7:(2,1),
    8:(0,2), 1:(1,2), 6:(2,2),
}
for n,(cx,cy) in pos.items():
    ax.text(cx+0.5, cy+0.5, f"{n} x{counts.get(n,0)}", ha='center', va='center')
ax.set_title("Lo-Shu Grid (counts)")
tmp = tempfile.mkstemp(suffix="_loshu.png")[1]
plt.tight_layout()
plt.savefig(tmp)
plt.close(fig)
return tmp

# ----- Report Builder ----- #
@dataclass
class Client:
    name: str
    dob: str # DD-MM-YYYY
    mobile: str
    lang: str # 'en' or 'hi'

def build_report_payload(c: Client, depth: str = 'detailed'):
    lp_comp, lp_root = life_path(c.dob)
    ex_comp, ex_root = pyth_total(c.name)
    su_comp, su_root = pyth_vowels(c.name)
    pe_comp, pe_root = pyth_consonants(c.name)
    bd_comp, bd_root = birthday_root(c.dob)
    mt_comp, mt_root = maturity_root(lp_root, ex_root)

    inc = inclusion_counts(c.name)
    missing = karmic_lessons(inc)
    hidden = hidden_passions(inc)
    bal = balance_number(c.name)
    kd = karmic_debts_from_all(c.dob, c.name)

    today = date.today()
    py = personal_year(c.dob, today)
    pm = personal_month(py, today.month)
    pdays = calendar_personal_days(c.dob, today)

    pins, pin_ages = pinnacles_and_ages(c.dob)
    chals = challenges(c.dob)

    mob_comp, mob_root = mobile_vibration(c.mobile) if c.mobile else (None,
None)

    base_inc = inc

```

```

candidates = generate_name_candidates(c.name)
scored = [score_candidate(x, lp_root, base_inc) for x in candidates]
scored.sort(key=lambda x: x['score'], reverse=True)

months = [(m, personal_month(py, m)) for m in range(1,13)]

return {
    'client': c,
    'depth': depth,
    'core': {
        'life_path': (lp_comp, lp_root),
        'expression': (ex_comp, ex_root),
        'soul': (su_comp, su_root),
        'personality': (pe_comp, pe_root),
        'birthday': (bd_comp, bd_root),
        'maturity': (mt_comp, mt_root),
    },
    'inclusion': inc,
    'missing': missing,
    'hidden': hidden,
    'balance': bal,
    'karmic_debts': kd,
    'py': py,
    'pm': pm,
    'pdays': pdays,
    'pinnacles': pins,
    'pin_ages': pin_ages,
    'challenges': chals,
    'mobile': (mob_comp, mob_root),
    'name_suggestions': scored,
    'months': months,
    'loshu': loshu_counts(c.dob),
}

# ----- PDF Class ----- #
DEJAVU_FONT_CANDIDATES = [
    "DejaVuSans.ttf",          # on Desktop next to numerology.py
    "/usr/share/fonts/truetype/dejavu/DejaVuSans.ttf",
    os.path.expanduser("~/Library/Fonts/DejaVuSans.ttf"),
]
DEVANAGARI_FONT_CANDIDATES = [
    "NotoSansDevanagari-Regular.ttf",
    "/Library/Fonts/NotoSansDevanagari-Regular.ttf",
    os.path.expanduser("~/Library/Fonts/NotoSansDevanagari-Regular.ttf"),
]

# ----- PDF class (must be ABOVE render_pdf) -----

```

```

class ReportPDF(FPDF):
    """Always use a Unicode font if available; never silently fall back."""
    def __init__(self, lang: str = "en"):
        super().__init__(orientation='P', unit='mm', format='A4')
        self.set_auto_page_break(auto=True, margin=15)
        self.lang = lang
        self.base_font = "Helvetica" # will be replaced by DejaVu/Deva if found
        self._load_fonts()

    @property
    def avail_w(self):
        return self.w - self.l_margin - self.r_margin

    def _left(self):
        self.set_x(self.l_margin)

    def _find_font(self, candidates):
        for p in candidates:
            if os.path.exists(p):
                return p
        return None

    def _load_fonts(self):
        # Find font files
        dejavu_path = self._find_font(DEJAVU_FONT_CANDIDATES)
        deva_path = self._find_font(DEVANAGARI_FONT_CANDIDATES)

        # Register fonts (fpdf2 ≥2.5 ignores "uni=")
        if dejavu_path:
            try: self.add_font("DejaVu", "", dejavu_path)
            except Exception: pass
        if deva_path:
            try: self.add_font("Deva", "", deva_path)
            except Exception: pass

        # Choose base font
        if self.lang == 'hi' and deva_path:
            self.base_font = "Deva"
        elif dejavu_path:
            self.base_font = "DejaVu"
        else:
            self.base_font = "Helvetica" # ASCII fallback

        # Set default font immediately so all writes use it
        self.set_font(self.base_font, "", 10)
        print("PDF base font selected:", self.base_font)

```

```

def _sanitize_if_needed(self, s: str) -> str:
    # If forced to Helvetica, downgrade Unicode punctuation to ASCII
    if self.base_font != "Helvetica":
        return s
    repl = {"—": "-", "·": ".", "“": "'", "”": "'", "„": "'", "“": "'", "”": "'", "...": "...", "\u00a0": " "}
    for k,v in repl.items(): s = s.replace(k,v)
    return s

def header(self):
    self._left()
    self.set_font(self.base_font, "", 14)
    title = "Numerology Report / अंक ज्योतिष रिपोर्ट" if self.lang=='hi' else
    "Numerology Report"
    self.cell(self.avail_w, 8, txt=self._sanitize_if_needed(title), ln=1, align='C')
    self.ln(2)

def h1(self, text):
    self._left()
    self.set_font(self.base_font, "", 13)
    self.set_text_color(20,20,20)
    self.cell(self.avail_w, 8, txt=self._sanitize_if_needed(str(text)), ln=1)

def h2(self, text):
    self._left()
    self.set_font(self.base_font, "", 11)
    self.set_text_color(40,40,40)
    self.cell(self.avail_w, 7, txt=self._sanitize_if_needed(str(text)), ln=1)

def para(self, text):
    self._left()
    self.set_text_color(0,0,0)
    self.set_font(self.base_font, "", 10)
    txt = "\n".join(textwrap.wrap(str(text), 90, break_long_words=True,
break_on_hyphens=True))
    self.multi_cell(self.avail_w, 5.5, txt=self._sanitize_if_needed(txt))
    self.ln(1)

def kv(self, key, val):
    self._left()
    self.set_font(self.base_font, "", 10)
    s = self._sanitize_if_needed(f"{key}: {val}")
    self.multi_cell(self.avail_w, 5.5, txt=s)

# ----- PDF rendering (must be BELOW the class) -----
def render_pdf(payload, save_path: str):

```

```

lang = payload['client'].lang
depth = payload.get('depth','detailed')
C = CONTENT_HI if lang=='hi' else CONTENT_EN

if FPDF is None:
    raise RuntimeError("fpdf2 not installed. Run: python3 -m pip install fpdf2")

pdf = ReportPDF(lang=lang)
pdf.add_page()

# Cover / Client block
pdf.h1("Client Details" if lang=='en' else "क्लाइंट विवरण")
nm = payload['client'].name
dob = payload['client'].dob
pdf.kv("Name" if lang=='en' else "नाम", nm)
pdf.kv("DOB", dob)
if payload['client'].mobile:
    pdf.kv("Mobile" if lang=='en' else "मोबाइल", payload['client'].mobile)
pdf.ln(2)
pdf.para(C["intro"])

# Core Numbers
pdf.h1("Core Numbers" if lang=='en' else "मुख्य अंक")
core = payload['core']
for label, key in ("Life Path","life_path"), ("Expression","expression"), ("Soul
Urge","soul"), ("Personality","personality"), ("Birthday","birthday"),
("Maturity","maturity"):
    comp, rootn = core[key]
    if key=="life_path":
        meaning = C.get("life_path", {}).get(rootn, "")
    elif key=="expression" and lang=='en':
        meaning = CONTENT_EN.get("expression",{}).get(rootn,"")
    elif key=="soul" and lang=='en':
        meaning = CONTENT_EN.get("soul",{}).get(rootn,"")
    elif key=="personality" and lang=='en':
        meaning = CONTENT_EN.get("personality",{}).get(rootn,"")
    else:
        meaning = ""
    lbl = label if lang=='en' else {"Life Path":"लाइफ
पाथ","Expression":"एक्सप्रेसन","Soul Urge":"सोल
अर्जी","Personality":"पर्सनालिटी","Birthday":"बर्थडे","Maturity":"मैच्योरिटी"}[label]
    pdf.h2(f"{lbl}: {rootn} (compound {comp})")
    if meaning: pdf.para(meaning)

# ...(keep the rest of your render_pdf exactly as you had it)...
# Remember to size images within page width:

```

```

# pdf.image(incl_img, w=min(170, pdf.avail_w))
# pdf.image(los_img, w=min(120, pdf.avail_w))

pdf.output(save_path)

# ----- PDF Rendering ----- #
def render_pdf(payload, save_path: str):
    lang = payload['client'].lang
    depth = payload.get('depth','detailed')
    C = CONTENT_HI if lang=='hi' else CONTENT_EN

    if FPDF is None:
        raise RuntimeError("fpdf2 not installed. Run: python3 -m pip install fpdf2")

    pdf = ReportPDF(lang=lang)
    pdf.add_page()

    # Cover / Client block
    pdf.h1("Client Details" if lang=='en' else "क्लाइंट विवरण")
    nm = payload['client'].name
    dob = payload['client'].dob
    pdf.kv("Name" if lang=='en' else "नाम", nm)
    pdf.kv("DOB", dob)
    if payload['client'].mobile:
        pdf.kv("Mobile" if lang=='en' else "मोबाइल", payload['client'].mobile)
    pdf.ln(2)
    pdf.para(C["intro"])

    # Core Numbers
    pdf.h1("Core Numbers" if lang=='en' else "मुख्य अंक")
    core = payload['core']
    for label, key in ("Life Path","life_path"), ("Expression","expression"), ("Soul
Urge","soul"), ("Personality","personality"), ("Birthday","birthday"),
("Maturity","maturity"):
        comp, rootn = core[key]
        if key=="life_path":
            meaning = C.get("life_path", {}).get(rootn, "")
        elif key=="expression" and lang=='en':
            meaning = CONTENT_EN.get("expression",{}).get(rootn,"")
        elif key=="soul" and lang=='en':
            meaning = CONTENT_EN.get("soul",{}).get(rootn,"")
        elif key=="personality" and lang=='en':
            meaning = CONTENT_EN.get("personality",{}).get(rootn,"")
        else:
            meaning = ""
        lbl = label if lang=='en' else {"Life Path":"लाइफ

```

```

पाथ","Expression":"एक्सप्रेसन","Soul Urge":"सोल
अर्जी","Personality":"पर्सनालिटी","Birthday":"बर्थडे","Maturity":"मैच्योरिटी"}[label]
    pdf.h2(f"{lbl}: {rootn} (compound {comp})")
    if meaning: pdf.para(meaning)

# Core narrative
pdf.add_page()
pdf.h1("Interpretation — Core Numbers" if lang=='en' else "व्याख्या — मुख्य
अंक")
for label, root, txt in make_core_narrative(payload, lang):
    pdf.h2(f"{label}: {root}")
    if txt: pdf.para(txt)

# Extended Life Path narrative + Expression explainer (EN)
if lang == 'en':
    lp_root = payload['core']['life_path'][1]
    long_lp = CONTENT_EN.get("life_path_long", {}).get(lp_root, "")
    if long_lp:
        pdf.add_page()
        pdf.h1("Deep Dive — Life Path")
        pdf.para(long_lp)

    pdf.h1("How You Execute — Expression")
    pdf.para(CONTENT_EN.get("expression_long",""))

# Planning model
pdf.add_page()
pdf.h1("How To Use Your Cycles" if lang=='en' else "अपने चक्रों का उपयोग कैसे
करें")
pdf.para(CONTENT_EN.get("cycles_planning","") if lang=='en'
    else "पर्सनल ईयर/मंथ/डे को कार्य-योजना के रूप में लें — 1/8 दिनों में पहल/निर्णय, 2/6
में सहयोग/सौंदर्य, 3/5 में संप्रेषण/परिवर्तन, 4/7 में संरचना/अध्ययन, 9 में समापन।")

# Inclusion + chart + narrative
pdf.add_page()
pdf.h1("Inclusion Chart & Traits" if lang=='en' else "इन्क्लूजन चार्ट और प्रवृत्तियाँ")
counts = payload['inclusion']
pdf.para(", ".join(f"{d}:{c}" for d,c in counts.items()))
miss = payload['missing']
hid = payload['hidden']
pdf.para(f"Missing numbers: {'', '.join(map(str,miss)) if miss else 'None'}" if
lang=='en'
    else f"मिसिंग अंक: {'', '.join(map(str,miss)) if miss else 'कोई नहीं'}")
if hid:
    pdf.para(f"Hidden Passion(s): {'', '.join(map(str,hid))}" if lang=='en'

```

```

        else f"हिडन पैशन: {' '.join(map(str, hid))}")
    pdf.para(f"Balance Number: {payload['balance']}" if lang=='en' else f"बैलेंस
नंबर: {payload['balance']}")
    try:
        incl_img = chart_inclusion(counts); pdf.image(incl_img, w=170);
os.remove(incl_img)
    except Exception:
        pass
    for tip in make_inclusion_narrative(payload, lang):
        pdf.para("• " + tip)

# Karmic Debts
pdf.add_page()
pdf.h1("Karmic Debts / Lessons" if lang=='en' else "कर्मिक ऋण/पाठ")
for line in make_karmic_debt_narrative(payload, lang):
    pdf.para("• " + line)

# Lo-Shu Grid
pdf.add_page()
pdf.h1("Lo-Shu Grid (DOB digits)" if lang=='en' else "लो-शु ग्रिड (जन्म तिथि)")
los = payload['loshu']
pdf.para(" ".join(f"{d}:{c}" for d,c in los.items()))
try:
    los_img = chart_loshu(los); pdf.image(los_img, w=120);
os.remove(los_img)
except Exception:
    pass
for tip in make_loshu_narrative(payload, lang):
    pdf.para("• " + tip)

# Pinnacles & Challenges
pdf.add_page()
pdf.h1("Pinnacles & Challenges Timeline" if lang=='en' else "पिनेकल्स और
चुनौतियाँ")
pins = payload['pinnacles']; ages = payload['pin_ages']
for i,num in enumerate(pins, start=1):
    a0,a1 = ages[i-1]
    pdf.h2(f"Pinnacle {i}: {num} — approx ages {a0}-{a1}")
    pdf.para("Opportunities and themes unfold here; align actions with this
tone." if lang=='en'
        else "यह अवधि अवसर और विषय लाती है; अपनी क्रियाओं को इसी स्वर के साथ
संरेखित करें।")
    ch = payload['challenges']
    pdf.h2("Challenges" if lang=='en' else "चुनौतियाँ")
    CMAP = CONTENT_EN['challenge_meanings'] if lang=='en' else

```



```

CONTENT_HI['challenge_meanings']
for i,cg in enumerate(ch, start=1):
    pdf.para(f"Challenge {i}: {cg} — {CMAP.get(cg, '')}" if lang=='en'
            else f"चुनौती {i}: {cg} — {CMAP.get(cg, '')}")

# Current cycles + predictions
pdf.add_page()
pdf.h1("Current Cycles" if lang=='en' else "वर्तमान चक्र")
today = date.today()
pdf.kv("Today" if lang=='en' else "आज", fmt_date(today))
pdf.kv("Personal Year" if lang=='en' else "पर्सनल ईयर", payload['py'])
YM = CONTENT_EN['year_meanings'] if lang=='en' else
CONTENT_HI['year_meanings']
if payload['py'] in YM: pdf.para(YM[payload['py']])
pdf.kv("Personal Month" if lang=='en' else "पर्सनल मंथ", payload['pm'])
pdf.para("Month tones refine the yearly theme. Plan accordingly." if
lang=='en'
      else "मासिक स्वर वार्षिक थीम को परिष्कृत करता है — योजना उसी अनुसार करें।")
pdf.h2("Actionable Outlook" if lang=='en' else "व्यावहारिक दृष्टिकोण")
for k,v in make_predictions(payload, lang).items():
    pdf.para(f"{k}: {v}")

# Personal Day Calendar (this month)
pdf.h2("This Month — Personal Day Calendar" if lang=='en' else "यह महीना —
पर्सनल डे कैलेंडर")
for idx,(d, pdn) in enumerate(payload['pdays'], start=1):
    pdf.para(f"{d.isoformat()}: {pdn}")
    if idx % 20 == 0:
        pdf.add_page()
        pdf.h2("Personal Day Calendar (cont.)" if lang=='en' else "पर्सनल डे कैलेंडर
(जारी)")

# 12-Month Planner
pdf.add_page()
pdf.h1("12-Month Personal Month Planner" if lang=='en' else "12-महीने का
व्यक्तिगत माह प्लानर")
YMAP = CONTENT_EN['year_meanings'] # short captions
for m, pm in payload['months']:
    pdf.para(f"Month {m}: {pm} — {YMAP.get(pm,'Cycle focus')}")

# Dasha-style timeline
pdf.add_page()
pdf.h1("Long-Term Dasha-Style Timeline (Numerology)" if lang=='en' else
"दीर्घकालीन दशा-शैली टाइमलाइन (न्यूमरोलॉजी)")
seq = [(1,6),(2,10),(3,16),(4,18),(5,17),(6,20),(7,7),(8,19),(9,7)]

```

```

start_num = payload['core']['life_path'][1]
idx = [n for n,_ in seq].index(reduce_num(start_num, True))
rot = seq[idx:]+seq[:idx]
age0 = 0
dasha_text = {
    1:"Sun-like initiative and self-definition.",
    2:"Moon-like partnership, patience, support.",
    3:"Jupiterian growth in learning, creativity, social reach.",
    4:"Work/structure tests; unconventional turns.",
    5:"Movement, change, communication spikes.",
    6:"Harmony: family, beauty, responsibilities, comfort.",
    7:"Introspection: study, spirituality, solitude.",
    8:"Power: management, duty, ethical achievement.",
    9:"Action and closure; courage to finish."
}
for n, yrs in rot:
    age1 = age0 + yrs
    pdf.h2(f"Number {n} period — approx ages {age0}-{age1}")
    pdf.para(dasha_text[n])
    age0 = age1

# Mobile number analysis
pdf.add_page()
pdf.h1("Mobile Number Analysis" if lang=='en' else "मोबाइल नंबर विश्लेषण")
mc, mr = payload['mobile']
if mr is not None:
    txt = f"Mobile vibration: {mr} (compound {mc}). Align with Life Path
{payload['core']['life_path'][1]} where possible; avoid debt compounds like
13/14/16/19."
    pdf.para(txt if lang=='en' else txt)
else:
    pdf.para("No mobile provided." if lang=='en' else "मोबाइल नंबर उपलब्ध नहीं।")

# Name correction suggestions
pdf.add_page()
pdf.h1("Name Correction Suggestions" if lang=='en' else "नाम सुधार सुझाव")
if payload['name_suggestions']:
    for i, it in enumerate(payload['name_suggestions'][:10], start=1):
        pdf.h2(f"{i}. {it['name']}")
        pdf.para(f"Score {it['score']} — PYT {it['pyt_root']} (comp
{it['pyt_compound']}), CH {it['chal_root']} (comp {it['chal_compound']}); covers
missing: {it['covered_missing']}")
    else:
        pdf.para("No suggestions generated — try adding a middle name/initial or
minor spelling change."
            if lang=='en' else "कोई सुझाव निर्मित नहीं — मध्य नाम/इनिशियल या छोटे वर्तनी
परिवर्तन आजमाएँ।")

```

```

# Remedies & Guidance
pdf.add_page()
pdf.h1("Remedies & Guidance" if lang=='en' else "उपाय व मार्गदर्शन")
pdf.para("Remedies aim to align mindset and environment with desired
energies. Adopt those that feel practical; consistency matters."
    if lang=='en' else
        "उपाय मनोवृत्ति और परिवेश को वांछित ऊर्जा के साथ संरेखित करने हेतु हैं — जो व्यावहारिक
        लगें, उन्हें नियमित रूप से अपनाएँ।")
    if payload['missing']:
        pdf.h2("For your missing numbers:" if lang=='en' else "आपके मिसिंग अंकों के
        लिए:")
        tips_map = {
            1:"Practice small acts of initiative daily; affirm 'I lead with wisdom'.",
            2:"Develop diplomacy; weekly acts of cooperation and patience.",
            3:"Schedule creative play; express through writing/speech.",
            4:"Build routines; use planners; one organized area at a time.",
            5:"Try new experiences safely; balance routine with variety.",
            6:"Nurture home ties; set healthy boundaries in caretaking.",
            7:"Meditate/journal; study; embrace quiet learning.",
            8:"Learn finance & leadership; act with integrity; Saturday charity.",
            9:"Practice forgiveness; monthly donation/service; finish loose ends."
        }
        for d in payload['missing']:
            pdf.para(f"{d}: {tips_map.get(d, 'Cultivate qualities of this digit
            consciously.')}")

# Appendix A: Number Archetypes (1–9)
pdf.add_page()
pdf.h1("Appendix A — Number Archetypes (1-9)" if lang=='en' else "परिशिष्ट A
— अंक आर्कटाइप (1-9)")
APP = CONTENT_EN['appendix_number'] if 'appendix_number' in
CONTENT_EN else {
    1:"1 — Initiative, autonomy, will.",
    2:"2 — Cooperation, rhythm, sensitivity.",
    3:"3 — Expression, art, optimism.",
    4:"4 — Order, practicality, systems.",
    5:"5 — Change, travel, sales, tech.",
    6:"6 — Harmony, service, beauty.",
    7:"7 — Research, spirituality, depth.",
    8:"8 — Management, finance, authority.",
    9:"9 — Compassion, global vision, finishing."
}
for n in range(1,10):
    pdf.h2(f"Number {n}")
    pdf.para(APP[n])

```

```

# Appendix B: Methodology & Checks
pdf.add_page()
pdf.h1("Appendix B — Methodology & Checks" if lang=='en' else "परिशिष्ट B — पद्धति")
pdf.para(
    "Pythagorean (A-Z→1-9) and Chaldean totals; Master 11/22/33 preserved at root. "
    "Inclusion chart tallies letter frequency 1–9; zeros are karmic lessons; tallest bars are hidden passions. "
    "Personal Year = day + month + current year (reduced); Month = PY + month; Day = PM + day. "
    "Pinnacles/Challenges per standard numerology formulae."
    if lang=='en' else
    "पायथागोरियन व काल्डियन योग; मास्टर 11/22/33 ज्यों के त्यों। "
    "इन्क्लूजन चार्ट में 1-9 की आवृत्ति; शून्य = कर्मिक पाठ; सबसे ऊँची आवृत्ति = हिडन पैशन। "
    "पर्सनल ईयर = दिन + माह + वर्ष (घटाकर); मंथ = PY + माह; डे = PM + दिन। "
    "पिनेकल/चुनौतियाँ मानक सूत्रों से। "
)

```

```

# Appendix C: Next Month Personal Day Calendar
next_month = date.today().replace(day=28) + timedelta(days=4)
first_of_next = next_month.replace(day=1)
pdf.add_page()
pdf.h1("Appendix C — Next Month Personal Day Calendar" if lang=='en' else "परिशिष्ट C — अगला महीना (पर्सनल डे)")
py2 = personal_year(payload['client'].dob, first_of_next)
pm2 = personal_month(py2, first_of_next.month)
days2 = calendar.monthrange(first_of_next.year, first_of_next.month)[1]
for d in range(1, days2+1):
    pdf.para(f"{date(first_of_next.year, first_of_next.month, d).isoformat()}: {personal_day(pm2, d)}")
    if d % 22 == 0 and d != days2:
        pdf.add_page()
        pdf.h2("Next Month (cont.)" if lang=='en' else "अगला महीना (जारी)")

```

```

# Depth padding summary
if depth == 'detailed':
    pdf.add_page()
    pdf.h1("Summary & Next Steps" if lang=='en' else "सारांश व आगे की राह")
    pdf.para("Use Personal Months/Days to time actions. Consider optimizing your name/mobile. Revisit this report each quarter; growth is cyclical."
        if lang=='en' else
        "क्रियाओं का समय निर्धारण करने हेतु पर्सनल मंथ/डे का उपयोग करें; नाम/मोबाइल

```

अनुकूलन पर विचार करें; प्रत्येक तिमाही में पुनरावलोकन करें।")

else:

pdf.add\_page()

pdf.h1("Summary" if lang=='en' else "सारांश")

pdf.para("Key insights captured above." if lang=='en' else "मुख्य अंतर्दृष्टियाँ  
ऊपर दी गई हैं।")

pdf.output(save\_path)

# ----- CSV Autosave ----- #

def autosave\_csv(payload):

headers = [

'timestamp','name','dob','mobile','lang','life\_path','expression','soul','personality',  
'birthday','maturity','py','pm'

]

row = {

'timestamp': datetime.now().isoformat(timespec='seconds'),

'name': payload['client'].name,

'dob': payload['client'].dob,

'mobile': payload['client'].mobile,

'lang': payload['client'].lang,

'life\_path': payload['core']['life\_path'][1],

'expression': payload['core']['expression'][1],

'soul': payload['core']['soul'][1],

'personality': payload['core']['personality'][1],

'birthday': payload['core']['birthday'][1],

'maturity': payload['core']['maturity'][1],

'py': payload['py'],

'pm': payload['pm']

}

try:

exists = os.path.exists(CSV\_PATH)

with open(CSV\_PATH, 'a', newline='') as f:

w = csv.DictWriter(f, fieldnames=headers)

if not exists: w.writeheader()

w.writerow(row)

except Exception:

pass

# ----- GUI ----- #

root = tk.Tk()

root.title(APP\_TITLE)

root.geometry("1120x800")

style = ttk.Style()

try:

```
style.theme_use('clam')
except Exception:
    pass
```

```
frm = ttk.Frame(root, padding=14)
frm.grid(row=0,column=0,sticky='nsew')
root.rowconfigure(0, weight=1)
root.columnconfigure(0, weight=1)
```

```
# Inputs
```

```
lbl_name = ttk.Label(frm, text="Full Name")
lbl_name.grid(row=0,column=0,sticky='w')
entry_name = ttk.Entry(frm, width=40)
entry_name.grid(row=1,column=0,sticky='we', padx=(0,10))
```

```
lbl_dob = ttk.Label(frm, text="DOB (DD-MM-YYYY)")
lbl_dob.grid(row=0,column=1,sticky='w')
entry_dob = ttk.Entry(frm, width=18)
entry_dob.grid(row=1,column=1,sticky='we')
```

```
lbl_mobile = ttk.Label(frm, text="Mobile (optional)")
lbl_mobile.grid(row=0,column=2,sticky='w')
entry_mobile = ttk.Entry(frm, width=20)
entry_mobile.grid(row=1,column=2,sticky='we', padx=(10,0))
```

```
lbl_lang = ttk.Label(frm, text="Language")
lbl_lang.grid(row=0,column=3,sticky='w')
lang_var = tk.StringVar(value='en')
combo_lang = ttk.Combobox(frm, textvariable=lang_var, values=['en','hi'],
width=8, state='readonly')
combo_lang.grid(row=1,column=3,sticky='w', padx=(10,0))
```

```
# Depth control
```

```
lbl_depth = ttk.Label(frm, text="Depth")
lbl_depth.grid(row=0, column=4, sticky='w')
depth_var = tk.StringVar(value='detailed')
combo_depth = ttk.Combobox(frm, textvariable=depth_var,
values=['concise','standard','detailed'], width=10, state='readonly')
combo_depth.grid(row=1, column=4, sticky='w', padx=(10,0))
```

```
btns = ttk.Frame(frm)
btns.grid(row=1,column=5,padx=(10,0))
```

```
# Preview notebook
```

```
nb = ttk.Notebook(frm)
nb.grid(row=2,column=0,columnspan=6,sticky='nsew', pady=(12,0))
frm.rowconfigure(2, weight=1)
```

```

frame_over = ttk.Frame(nb); nb.add(frame_over, text="Overview")
sbar_over = ttk.Scrollbar(frame_over, orient='vertical')
txt_over = tk.Text(frame_over, wrap='word', yscrollcommand=sbar_over.set)
sbar_over.config(command=txt_over.yview)
sbar_over.pack(side='right', fill='y'); txt_over.pack(side='left', fill='both',
expand=True)

```

```

_payload = None

```

```

def do_generate():
    name = entry_name.get().strip()
    dob = entry_dob.get().strip()
    mobile = entry_mobile.get().strip()
    lang = lang_var.get()
    depth = depth_var.get()
    if not name or not dob:
        messagebox.showerror("Missing", "Please enter Full Name and DOB.")
        return
    try:
        datetime.strptime(dob, "%d-%m-%Y")
    except ValueError:
        messagebox.showerror("Date error", "Use DD-MM-YYYY.")
        return
    c = Client(name=name, dob=dob, mobile=mobile, lang=lang)
    payload = build_report_payload(c, depth=depth)
    # preview
    txt_over.configure(state='normal'); txt_over.delete('1.0','end')
    core = payload['core']
    txt_over.insert('end', f"Name: {name}\nDOB: {dob}\n\n")
    for label,key in ("Life Path","life_path"),("Expression","expression"),("Soul
Urge","soul"),("Personality","personality"),("Birthday","birthday"),
("Maturity","maturity"):
        comp,rootn = core[key]
        txt_over.insert('end', f"{label}: {rootn} (compound {comp})\n")
    txt_over.insert('end', f"Personal Year: {payload['py']}\nPersonal Month:
{payload['pm']}\n")
    txt_over.insert('end', f"Missing: {payload['missing']}\nHidden Passions:
{payload['hidden']}\n")
    txt_over.insert('end', f"Language: {lang} | Depth: {depth}\n")
    txt_over.configure(state='disabled')
    global _payload; _payload = payload
    autosave_csv(payload)

```

```

def do_export_pdf():
    if not _payload:
        messagebox.showerror("No report", "Generate a report first.")

```

```

        return
    if _payload['client'].lang=='hi':
        found = any(os.path.exists(p) for p in DEVANAGARI_FONT_CANDIDATES)
        if not found:
            messagebox.showerror(
                "Hindi font missing",
                "Place NotoSansDevanagari-Regular.ttf next to numerology.py (or
update DEVANAGARI_FONT_CANDIDATES)."
            )
            return
        initial = os.path.join(DEFAULT_SAVE_DIR,
f"{_payload['client'].name.replace(' ','_')}_Numerology_Report.pdf")
        path = filedialog.asksaveasfilename(defaultextension='.pdf',
initialfile=os.path.basename(initial), initialdir=DEFAULT_SAVE_DIR,
filetypes=[('PDF','*.pdf')])
        if not path: return
        try:
            render_pdf(_payload, path)
            messagebox.showinfo("Saved", f"PDF saved to: {path}")
        except Exception as e:
            messagebox.showerror("Save failed", f"Could not save PDF:\n{e}")

btn_gen = ttk.Button(btns, text="Generate Report", command=do_generate)
btn_pdf = ttk.Button(btns, text="Export PDF", command=do_export_pdf)
btn_gen.grid(row=0,column=0,padx=(0,8))
btn_pdf.grid(row=0,column=1)

for t in (txt_over,):
    t.configure(state='normal')
    t.insert('end', 'Fill details above → Generate Report → Export PDF')
    t.configure(state='disabled')

root.mainloop()

```