**🧩 Enhanced Architecture**

**🏠 1. Onboarding & Account Setup**

Purpose: introduce users into storyworld and initialize game data.  
Key Additions: first tutorial quest + default skill trees + difficulty presets.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 1.1 Welcome Screen | Short story intro (“You awaken in the Land of Growth…”) | — |
| 1.2 Login / Signup | Email / OAuth / guest mode + cloud sync token | Users |
| 1.3 Character Creation | Choose avatar, name, goals, starter class | Characters, Skills |
| 1.4 Class Quiz | Personality assessment → auto‑assign class | Users(metadata) |
| 1.5 Story Tutorial Quest | First “quest” explaining XP, HP, Coins | Projects, Tasks |
| 1.6 Sync & Permissions | Notifications, backup, offline setup | Settings |

**⚙️ 2. Settings & Game Rules**

Purpose: tailor difficulty, frequency, and data control.  
Key Additions: preset modes (Easy / Normal / Hard) + theme switcher.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 2.1 Game Settings | XP/HP gain rates + streak bonuses | Settings |
| 2.2 Scaling & Prestige | Level curve formula + prestige toggle | Settings |
| 2.3 Economy Rules | Coin rates + overdraft penalties + hotel ratio | Settings |
| 2.4 Notification | Reminder times + channels | Settings |
| 2.5 Account | Change email, password, avatar | Users |
| 2.6 Data & Sync | Backup/export + sync devices | Settings |

**💪 3. Habits & Skills**

Purpose: anchor daily growth + track streaks and mastery.  
Key Additions: habit templates library + accountability buddy + passive bonuses.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 3.1 Habits Dashboard | View/manage all habits (good + bad) | Habits, Skills |
| 3.2 Add / Edit Habit | Create habit, assign skill, set XP value | Habits |
| 3.3 Habit Detail View | Graphs, streaks, stats | Habits logs, Events |
| 3.4 Bad Habit Battle | Visual fight animation → HP drops | Events, HabitBattles |
| 3.5 Skill Tree Overview | Visual map + unlock logic | Skills, Achievements |
| 3.6 Skill Detail | Perks + passive buffs | Skills, Achievements |
| 3.7 Habit Template Library | Pre‑made habits (by skill/goal) | Templates |

**🧠 4. PARA Productivity System**

Purpose: transform tasks → quests under skill domains.  
Key Additions: quest deadline calendar + XP summary card + auto reward pop‑up.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 4.1 Brain Dump | Quick ideas inbox | Tasks |
| 4.2 PARA Dashboard | Overview of Projects/Areas/Resources/Archive | Projects, Areas |
| 4.3 Project (Quest) | Description + difficulty + deadline | Projects, Tasks |
| 4.4 Task (Subquest) Detail | Checklist, XP rewards | Tasks |
| 4.5 Resource Library | Knowledge / notes | Resources |
| 4.6 Archive / Logs | Completed quests / XP records | Archive, Events |
| 4.7 Quest Calendar | Timeline / due dates | Tasks(deadline) |

**💰 5. Economy & Marketplace**

Purpose: manage rewards and spending loops.  
Key Additions: rarity tiers + daily bonus chest + inventory fusion + trading.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 5.1 Marketplace | Buy boosts, skins, avatars | Items, Transactions |
| 5.2 Black Market | Risk items / chance rolls | Items, RNG table |
| 5.3 Hotel / Recovery | Spend coins → restore HP | Users, Transactions |
| 5.4 Inventory & Fusion | Combine items / view collection | Inventory |
| 5.5 Transactions Log | Earn / spend history | Transactions |
| 5.6 Daily Reward Chest | Randomized item drop once per day | Events, RNG table |

**🔮 6. AI Companion**

Purpose: personalize experience + reflect progress.  
Key Additions: selectable tone/personality + mission cooldowns.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 6.1 Companion Home | Chat interface | AI\_Logs |
| 6.2 Daily Missions | Auto‑generated quests | Tasks, AI\_Logs |
| 6.3 Weekly Report | Narrative summary + trend graphs | Events, Skills |
| 6.4 Reflection Journal | Write entries → Wisdom XP | Journal |
| 6.5 Mood Analysis | Sentiment → advice | AI\_Logs |
| 6.6 Recommendations | Suggested next steps | AI\_Logs |

**💬 7. Community & Social**

Purpose: cooperative play and competitive motivation.  
Key Additions: guild XP pools + party quests + seasonal events + moderation tools.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 7.1 Guilds Hub | Join/create guilds | Guilds |
| 7.2 Guild Dashboard | Shared goals & XP | Guilds, Guild\_Members |
| 7.3 Guild Chat | Real‑time messaging | Messages |
| 7.4 Leaderboards | Top XP, skills, streaks | Users, Achievements |
| 7.5 Friend List | View friends, co‑quests | Friends |
| 7.6 Party Quests | Cooperative projects | Projects, Party |
| 7.7 Mentorship Hub | Mentor–mentee links | Mentorship |
| 7.8 Seasonal Events | Rotating global quests | Events, RNG table |

**🧾 8. Visualization & Dashboard**

Purpose: show progress + game feedback.  
Key Additions: weekly reflection pop‑up + cut‑scene level‑ups.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 8.1 Home Dashboard | Character card + quick actions | Users, Skills |
| 8.2 XP & Analytics | Graphs / charts | Events |
| 8.3 Recent Logs | XP gained, damage, coins | Events, Transactions |
| 8.4 Event Cards | Random daily good/bad events | RNG table |
| 8.5 Prestige Page | Rebirth stats + bonuses | Users, Achievements |

**🛡️ 9. Backend / System Management**

Purpose: admin tools + scheduled functions (n8n ready).

| **Screen / Function** | **Description** | **Backend Links** |
| --- | --- | --- |
| 9.1 Cron Jobs | Weekly HP deductions + event generation | System Logs |
| 9.2 Balance Engine | Global XP/HP adjustments | Settings |
| 9.3 Event Generator | Create daily RNG events | RNG table |
| 9.4 Data Admin | Edit stats + backup + analytics | Admin, System Logs |

**🧩 10. Meta / Global Pages**

Purpose: utilities and lore.  
Key Additions: dark mode / retro theme and achievement showcase.

| **Screen** | **Description** | **Backend Links** |
| --- | --- | --- |
| 10.1 Notifications Center | Central alerts | Events |
| 10.2 Help / Tutorials | Guides & FAQ | Resources |
| 10.3 Credits & Lore | Worldbuilding / timeline | Lore |
| 10.4 Achievements Hall | Unlocked titles + rewards | Achievements |
| 10.5 Legal / Privacy | ToS / delete account | Users |

**🗺️ SYSTEM–PAGE MAPPING DOCUMENT**

Below shows how front‑end pages connect to backend tables + automations (like n8n flows).

| **Page / Screen** | **DB Tables Used** | **n8n / Automation Flow (Triggered or Referenced)** |
| --- | --- | --- |
| Character Creation | Users, Characters, Skills | INIT\_USER\_SETUP → creates default skills + tutorial quest |
| Story Tutorial Quest | Projects, Tasks | QUEST\_AUTOSTART → marks first quest complete for onboarding |
| Habits Dashboard | Habits, Skills, Events | HABIT\_CHECKIN → XP update, streak bonus, HP loss calc |
| Skill Tree Overview | Skills, Achievements | SKILL\_SYNC → update visual map & unlock perks |
| Habit Battle | Events (HabitBattles) | DAMAGE\_CALC → HP subtract → Triggers visual animation |
| Projects / Tasks | Projects, Tasks, Events | QUEST\_ENGINE → reward distribution / difficulty modifier |
| Quest Calendar | Tasks(deadline), Events | REMINDER\_FLOW → daily telegram or notif alerts |
| Marketplace | Items, Transactions | SHOP\_CHECK → coin validation, item grant |
| Black Market | Items, RNG table | RANDOM\_EVENT\_ROLL → success / failure effect |
| Hotel Recovery | Users, Transactions | HP\_RESTORE → coin deduct / HP restore |
| Inventory Fusion | Inventory, Items | ITEM\_COMBINE → creates new item + logs Event |
| AI Companion | AI\_Logs, Tasks | AI\_MISSIONS → GPT prompt based on stats history |
| Reflection Journal | Journal, Events | WISDOM\_XP\_UPDATE → add XP per entry |
| Guild Dashboard | Guilds, Guild\_Members | GUILD\_SYNC → share XP pool, update leaderboard |
| Party Quests | Projects, Party | PARTY\_REWARD → split XP among members |
| Leaderboards | Users, Achievements | LEADERBOARD\_REFRESH (cron weekly) |
| Community Chat | Messages, Users | SOCKET\_HANDLER → real‑time updates |
| Seasonal Events | Events, RNG table | EVENT\_SEEDER (cron start of month) |
| Home Dashboard | Users, Skills, Events | DASHBOARD\_RENDER → fetch aggregate stats |
| Prestige Page | Users, Achievements | PRESTIGE\_CALC → reset XP ☞ add bonus |
| Notifications | Events, Settings | ALERT\_ENGINE → mobile push / email |
| Achievements Hall | Achievements, Inventory | ACHIEVEMENT\_UNLOCK → trigger badge + XP bonus |
| Cron Jobs Dashboard | System Logs / Events | CRON\_MANAGER → trigger maintenance flows |
| Event Generator | RNG table, Events | RNG\_MIX → generate new event cards daily |

**🧭 Data Diagram Perspective (simplified)**

text

Users ─┬─ Characters ─┬─ Skills ─┬─ Habits

│ │ └─ Events

│ └─ Projects ─┬─ Tasks

│ └─ Transactions

├─ Inventory ─ Items ─ RNG table

├─ Guilds ─ Guild\_Members ─ Party

├─ Achievements

├─ AI\_Logs (GPT missions)

└─ Settings / System\_Logs

**🧮 n8n Automation Suite (Backbone)**

| **Flow Name** | **Trigger** | **Outcome** |
| --- | --- | --- |
| INIT\_USER\_SETUP | On new user signup | Creates character + skills + tutorial quest |
| HABIT\_CHECKIN | Manual habit mark done | Adds XP/coins + updates skill XP |
| DAMAGE\_CALC | When bad habit marked | Reduces HP + logs event |
| QUEST\_ENGINE | Task completion | Gives XP + coins + streak multiplier |
| SHOP\_CHECK | Item purchase | Deducts coins, adds item to inventory |
| CRON\_MANAGER | Time schedule | Runs HP penalty / bonus / daily events |
| AI\_MISSIONS | Daily | Generates missions via AI Companion |
| ACHIEVEMENT\_UNLOCK | XP threshold crossed | Grants title + badge |
| EVENT\_SEEDER | Monthly | Adds new random events into pool |
| PRESTIGE\_CALC | Level max | Resets stats / adds permanent bonus |

**5. Example User Flow**

1. User creates account → chooses avatar → fills in goals
2. App generates default skill trees (Health, Work, Finance)
3. User adds habits → starts daily tracking
4. Completing habits earns XP + coins
5. User buys “rewards” in market → upgrades avatar
6. If user overspends → overdraft penalty reduces HP weekly
7. User interacts with others in the guild chat
8. AI companion gives feedback like “You’re 200 XP away from next level!”

**🏦 SQL Database Tables & Relationships Life Game**

**1. USERS**

sql

**CREATE** **TABLE** users (

id **SERIAL** **PRIMARY** **KEY**,

email **VARCHAR**(255) **UNIQUE** NOT NULL,

username **VARCHAR**(80) **UNIQUE** NOT NULL,

avatar **VARCHAR**(255),

join\_date **TIMESTAMP** **DEFAULT** NOW(),

password\_hash **VARCHAR**(255),

theme **VARCHAR**(40) **DEFAULT** 'default'

);

**2. CHARACTERS**

sql

**CREATE** **TABLE** characters (

id **SERIAL** **PRIMARY** **KEY**,

user\_id **INTEGER** **REFERENCES** users(id),

class **VARCHAR**(32),

bio **TEXT**,

goals **TEXT**,

**level** **INTEGER** **DEFAULT** 1,

xp **INTEGER** **DEFAULT** 0,

hp **INTEGER** **DEFAULT** 100,

coins **INTEGER** **DEFAULT** 100,

prestige\_level **INTEGER** **DEFAULT** 0,

title **VARCHAR**(120),

last\_login **TIMESTAMP**

);

**3. SKILLS**

sql

**CREATE** **TABLE** skills (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

name **VARCHAR**(64),

xp **INTEGER** **DEFAULT** 0,

**level** **INTEGER** **DEFAULT** 1,

unlocked **BOOLEAN** **DEFAULT** FALSE

);

**4. HABITS**

sql

**CREATE** **TABLE** habits (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

skill\_id **INTEGER** **REFERENCES** skills(id),

name **VARCHAR**(100),

**type** **VARCHAR**(10) **CHECK** (**type** IN ('good','bad')),

xp\_value **INTEGER** **DEFAULT** 0,

hp\_value **INTEGER** **DEFAULT** 0,

streak **INTEGER** **DEFAULT** 0,

last\_completed **DATE**,

template\_id **INTEGER** **REFERENCES** habit\_templates(id)

);

**5. HABIT\_TEMPLATES**

sql

**CREATE** **TABLE** habit\_templates (

id **SERIAL** **PRIMARY** **KEY**,

name **VARCHAR**(80),

skill\_name **VARCHAR**(64),

description **TEXT**

);

**6. EVENTS**

sql

**CREATE** **TABLE** events (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

event\_type **VARCHAR**(50),

value\_change **INTEGER**,

xp\_change **INTEGER**,

hp\_change **INTEGER**,

coins\_change **INTEGER**,

description **TEXT**,

event\_date **TIMESTAMP** **DEFAULT** NOW()

);

**7. PROJECTS (QUESTS)**

sql

**CREATE** **TABLE** projects (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

area\_id **INTEGER** **REFERENCES** areas(id),

title **VARCHAR**(120),

description **TEXT**,

total\_xp **INTEGER** **DEFAULT** 0,

coin\_reward **INTEGER** **DEFAULT** 0,

difficulty **VARCHAR**(32),

deadline **DATE**,

completed **BOOLEAN** **DEFAULT** FALSE

);

**8. AREAS (PARA Structure)**

sql

**CREATE** **TABLE** areas (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

name **VARCHAR**(50),

description **TEXT**

);

**9. TASKS (SUBQUESTS)**

sql

**CREATE** **TABLE** tasks (

id **SERIAL** **PRIMARY** **KEY**,

project\_id **INTEGER** **REFERENCES** projects(id),

title **VARCHAR**(120),

completed **BOOLEAN** **DEFAULT** FALSE,

xp **INTEGER** **DEFAULT** 0,

coins **INTEGER** **DEFAULT** 0,

difficulty **VARCHAR**(32)

);

**10. RESOURCES (Knowledge Notes)**

sql

**CREATE** **TABLE** resources (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

title **VARCHAR**(120),

**type** **VARCHAR**(32),

content **TEXT**,

added **TIMESTAMP** **DEFAULT** NOW()

);

**11. ARCHIVE / LOGS**

sql

**CREATE** **TABLE** archive (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

project\_id **INTEGER** **REFERENCES** projects(id),

completed\_on **TIMESTAMP**,

xp\_earned **INTEGER**,

coins\_earned **INTEGER**

);

**12. ITEMS (Marketplace/Inventory)**

sql

**CREATE** **TABLE** items (

id **SERIAL** **PRIMARY** **KEY**,

name **VARCHAR**(64),

item\_type **VARCHAR**(32),

rarity **VARCHAR**(32),

description **TEXT**,

effect **TEXT**,

cost **INTEGER** **DEFAULT** 0

);

**13. INVENTORY**

sql

**CREATE** **TABLE** inventory (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

item\_id **INTEGER** **REFERENCES** items(id),

quantity **INTEGER** **DEFAULT** 1,

acquired **TIMESTAMP** **DEFAULT** NOW()

);

**14. TRANSACTIONS**

sql

**CREATE** **TABLE** **transactions** (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

**type** **VARCHAR**(32), *-- 'earn', 'spend', 'penalty', 'restore'*

amount **INTEGER**,

item\_id **INTEGER** **REFERENCES** items(id),

description **TEXT**,

trans\_date **TIMESTAMP** **DEFAULT** NOW()

);

**15. GUILDS**

sql

**CREATE** **TABLE** guilds (

id **SERIAL** **PRIMARY** **KEY**,

name **VARCHAR**(100) **UNIQUE**,

description **TEXT**,

leader\_id **INTEGER** **REFERENCES** users(id),

xp\_pool **INTEGER** **DEFAULT** 0,

created **TIMESTAMP** **DEFAULT** NOW()

);

**16. GUILD\_MEMBERS**

sql

**CREATE** **TABLE** guild\_members (

guild\_id **INTEGER** **REFERENCES** guilds(id),

user\_id **INTEGER** **REFERENCES** users(id),

joined **TIMESTAMP** **DEFAULT** NOW(),

is\_admin **BOOLEAN** **DEFAULT** FALSE,

**PRIMARY** **KEY**(guild\_id, user\_id)

);

**17. PARTY**

sql

**CREATE** **TABLE** party (

id **SERIAL** **PRIMARY** **KEY**,

project\_id **INTEGER** **REFERENCES** projects(id),

guild\_id **INTEGER** **REFERENCES** guilds(id),

**status** **VARCHAR**(32)

);

**18. FRIENDS**

sql

**CREATE** **TABLE** friends (

user\_id **INTEGER** **REFERENCES** users(id),

friend\_id **INTEGER** **REFERENCES** users(id),

**status** **VARCHAR**(32) **DEFAULT** 'pending',

**PRIMARY** **KEY**(user\_id, friend\_id)

);

**19. MESSAGES (Community, Guild Chat)**

sql

**CREATE** **TABLE** messages (

id **SERIAL** **PRIMARY** **KEY**,

sender\_id **INTEGER** **REFERENCES** users(id),

guild\_id **INTEGER** **REFERENCES** guilds(id),

content **TEXT**,

sent **TIMESTAMP** **DEFAULT** NOW()

);

**20. ACHIEVEMENTS**

sql

**CREATE** **TABLE** achievements (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

title **VARCHAR**(120),

description **TEXT**,

reward\_type **VARCHAR**(32),

bonus\_value **INTEGER**,

unlocked\_on **TIMESTAMP** **DEFAULT** NOW()

);

**21. SETTINGS**

sql

**CREATE** **TABLE** settings (

user\_id **INTEGER** **REFERENCES** users(id),

level\_xp\_formula **TEXT**,

overdraft\_rule **TEXT**,

notification\_times **TEXT**,

theme **VARCHAR**(32),

**PRIMARY** **KEY**(user\_id)

);

**22. JOURNAL (Reflection, AI)**

sql

**CREATE** **TABLE** journal (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

entry **TEXT**,

wisdom\_xp **INTEGER** **DEFAULT** 0,

entry\_date **TIMESTAMP** **DEFAULT** NOW()

);

**23. AI\_LOGS**

sql

**CREATE** **TABLE** ai\_logs (

id **SERIAL** **PRIMARY** **KEY**,

character\_id **INTEGER** **REFERENCES** characters(id),

message **TEXT**,

insight\_type **VARCHAR**(32),

**timestamp** **TIMESTAMP** **DEFAULT** NOW()

);

**24. RNG\_EVENTS (Random Cards, Daily Events)**

sql

**CREATE** **TABLE** rng\_events (

id **SERIAL** **PRIMARY** **KEY**,

description **TEXT**,

effect **TEXT**,

rarity **VARCHAR**(32),

available **BOOLEAN** **DEFAULT** TRUE,

last\_issued **DATE**

);

**25. MENTORSHIP**

sql

**CREATE** **TABLE** mentorship (

id **SERIAL** **PRIMARY** **KEY**,

mentor\_id **INTEGER** **REFERENCES** users(id),

mentee\_id **INTEGER** **REFERENCES** users(id),

started **TIMESTAMP** **DEFAULT** NOW(),

**status** **VARCHAR**(32)

);

**25. Systems Log**

sql

**CREATE** **TABLE** systems\_log (

id **SERIAL** **PRIMARY** **KEY**,

**timestamp** **TIMESTAMP** **DEFAULT** NOW(), *-- when the log was written*

actor\_type **VARCHAR**(40), *-- 'user', 'system', 'cron', 'admin', 'guild', etc.*

actor\_id **INTEGER**, *-- references users.id, admin id, automation id, etc.*

target\_type **VARCHAR**(40), *-- e.g. 'character', 'habit', 'project', 'guild', etc.*

target\_id **INTEGER**, *-- references target entity id*

**action** **VARCHAR**(100), *-- short action description ('habit\_marked', 'quest\_completed', 'coin\_spent', etc.)*

detail **TEXT**, *-- full description or JSONB detail for context*

outcome **VARCHAR**(64), *-- result or status ('success', 'failure', 'warning', etc.)*

severity **VARCHAR**(20) **DEFAULT** 'info', *-- info / warning / error / critical*

source **VARCHAR**(80), *-- where did it originate? ('web', 'mobile', 'n8n', 'system\_cron', etc.)*

ip\_address **VARCHAR**(64), *-- optional, for debugging/security tracing*

extra JSONB **DEFAULT** '{}' *-- extensible field for future metadata*

);

**🔗 ALL CONNECTIONS AND RELATIONS**

* Each user has one character (1:1)
* Each character has many skills, habits, achievements, inventory, journal, archive, projects, areas, resources, and events
* Each habit belongs to a skill
* Each project (quest) belongs to an area and has many tasks
* Guilds have many guild\_members; each user can join many guilds (M:N via guild\_members)
* Party links quests/projects and guilds together (for coop quests)
* Inventory tracks item ownership per character
* Friendship table links two users (M:N)
* Messages belong to a sender (user) and have an optional guild id (for chat rooms)
* Settings are per user
* AI logs and journals are per character for companion/reflective interactions
* Mentorship connects mentors and mentees by user id
* Events and rng\_events track all status changes, battles, bonuses, etc.

**🔗 Logging**

* Log every backend automation step: When an n8n workflow triggers (habit XP update, HP penalty, coin shop purchase), write an entry.
* Track user/session activity: When users complete quests/habits or change key settings.
* Monitor admin/system ops: When cron jobs or admin actions (e.g., data reset, mass notifications) run.
* Debugging and security: Records IP, timestamp, and relevant details so you can trace bugs, suspicious activity, or audit gameplay flows.
* Extensible "extra" field: Allows for future logging needs; e.g., storing diff snapshots, additional flags, etc.

**🔗 Relation Advice**

* The actor\_type and actor\_id allow you to link log activity to any other table/entity.
* Use indexes on timestamp, action, and outcome for fast querying/filtering, especially when scaling logs for analytics or audits.

**N8N Flows**

**1️⃣ INIT\_USER\_SETUP**

**Trigger:** New user signup (Users table)  
**Purpose:** Create character, assign skills, setup tutorial quest

**Steps:**

1. **Trigger** → New user created
2. **Create Character** → Characters table
3. **Assign Default Skills** → Skills table
4. **Initialize Tutorial Quest** → Projects / Tasks table
5. **Setup Default Settings** → Settings table
6. **Create Inventory / Starter Items** → Inventory table (optional)
7. **Generate Cloud Sync Token** → Users table update
8. **Log Onboarding Event** → Events table
9. **Trigger Follow-up Flows** → QUEST\_AUTOSTART, DASHBOARD\_RENDER

**Reusable Modules:**

* **Skill Assignment** → Used by INIT\_USER\_SETUP, QUEST\_ENGINE (for XP updates)
* **Event Logging** → Can be used by any flow (HABIT\_CHECKIN, DAMAGE\_CALC, QUEST\_ENGINE, etc.)

**2️⃣ HABIT\_CHECKIN**

**Trigger:** Manual habit marked done (Habits table)  
**Purpose:** Update XP, coins, and skill XP

**Steps:**

1. **Trigger** → Habit marked done
2. **Calculate Rewards** → XP, coins, streak multiplier
3. **Update Habit / Skill XP** → Habits + Skills table
4. **Log Event** → Events table

**Reusable Modules:**

* **Reward Calculation Module** → Can be reused in QUEST\_ENGINE, ACHIEVEMENT\_UNLOCK
* **Event Logging Module** → Already used in INIT\_USER\_SETUP

**3️⃣ DAMAGE\_CALC**

**Trigger:** Bad habit marked (HabitBattles / Events)  
**Purpose:** Reduce HP, log damage

**Steps:**

1. **Trigger** → Bad habit completed
2. **Calculate Damage** → HP deduction (possibly with skill modifiers)
3. **Update Character HP** → Users / Characters table
4. **Log Event** → Events table
5. **Trigger Visual Animation** → optional front-end call

**Reusable Modules:**

* **Event Logging** → Already shared
* **HP Adjustment Module** → Can be reused in CRON\_MANAGER, HOTEL\_RESTORE

**4️⃣ QUEST\_ENGINE**

**Trigger:** Task / Project completion (Tasks / Projects)  
**Purpose:** Grant XP, coins, streak bonuses

**Steps:**

1. **Trigger** → Task marked complete
2. **Calculate Rewards** → XP, coins, streak multiplier
3. **Update Skills** → Increment skill XP
4. **Update User Stats** → Users / Characters table
5. **Check for Achievement Thresholds** → call ACHIEVEMENT\_UNLOCK
6. **Log Event** → Events table

**Reusable Modules:**

* **Reward Calculation Module** → Shared with HABIT\_CHECKIN
* **Event Logging Module** → Shared
* **Skill Update Module** → Shared

**5️⃣ SHOP\_CHECK**

**Trigger:** Item purchase (Items / Transactions)  
**Purpose:** Deduct coins, add item to inventory

**Steps:**

1. **Trigger** → Purchase request
2. **Validate Coins** → Check Users / Characters table
3. **Deduct Coins** → Update Users / Characters
4. **Add Item to Inventory** → Inventory table
5. **Log Transaction Event** → Transactions / Events

**Reusable Modules:**

* **Transaction Logger** → Could feed into ACHIEVEMENT\_UNLOCK (wealth milestones)
* **HP / XP Modifier Module** → Only if items give buffs

**6️⃣ CRON\_MANAGER**

**Trigger:** Scheduled (daily/weekly)  
**Purpose:** Run HP penalties, bonuses, generate events

**Steps:**

1. **Trigger Cron** → Daily / Weekly schedule
2. **Apply HP / XP Updates** → Users / Characters table
3. **Generate Daily Events** → call EVENT\_SEEDER or RNG\_MIX
4. **Log Event** → Events table

**Reusable Modules:**

* **HP Adjustment Module** → shared with DAMAGE\_CALC
* **Event Seeder Module** → shared with EVENT\_SEEDER
* **Reward Calculation Module** → optional, if daily bonuses exist

**7️⃣ AI\_MISSIONS**

**Trigger:** Daily (AI\_Logs)  
**Purpose:** Generate missions via AI companion

**Steps:**

1. **Trigger** → Daily AI mission cycle
2. **Fetch User Stats / Skills** → Users, Skills, Events
3. **Generate Missions** → AI API call (GPT prompt)
4. **Insert New Tasks / Projects** → Tasks / Projects table
5. **Log Event** → AI\_Logs

**Reusable Modules:**

* **Event Logging Module** → Shared
* **Task Creation Module** → Reusable with QUEST\_ENGINE

**8️⃣ ACHIEVEMENT\_UNLOCK**

**Trigger:** XP / milestone threshold crossed (Users / Skills)  
**Purpose:** Grant title, badge, and bonus

**Steps:**

1. **Trigger** → XP threshold crossed
2. **Grant Achievement** → Achievements / Inventory table (badge/reward)
3. **Update User Stats** → XP/bonus if applicable
4. **Log Event** → Events table

**Reusable Modules:**

* **Event Logging Module** → Shared
* **Reward Calculation Module** → Shared

**9️⃣ EVENT\_SEEDER**

**Trigger:** Monthly (Events / RNG\_table)  
**Purpose:** Add new random events into the pool

**Steps:**

1. **Trigger Cron** → Monthly
2. **Generate Random Events** → RNG logic / RNG\_table
3. **Insert Events** → Events table
4. **Notify Users** → optional push / dashboard update
5. **Log Event** → Events table

**Reusable Modules:**

* **RNG Event Generator** → Shared with CRON\_MANAGER (daily events)
* **Event Logging Module** → Shared

**🔟 PRESTIGE\_CALC**

**Trigger:** Level max reached (Users / Characters)  
**Purpose:** Reset stats, add permanent bonus

**Steps:**

1. **Trigger** → Level threshold reached
2. **Calculate Prestige Bonus** → XP/HP/permanent perks
3. **Reset User Stats** → XP/HP/Skills to base
4. **Update Achievements / Titles** → optional
5. **Log Event** → Events table

**Reusable Modules:**

* **Reward Calculation Module** → Shared
* **Event Logging Module** → Shared
* **Skill Reset Module** → Could be shared with INIT\_USER\_SETUP for default skill setup

**💡 Reusable Modules Across Flows**

1. **Reward Calculation Module** → XP, coins, streaks
2. **Event Logging Module** → All XP, HP, item, mission updates
3. **HP / XP Modifier Module** → DAMAGE\_CALC, CRON\_MANAGER, PRESTIGE\_CALC
4. **Skill Update Module** → HABIT\_CHECKIN, QUEST\_ENGINE, PRESTIGE\_CALC
5. **Task / Project Creation Module** → INIT\_USER\_SETUP, AI\_MISSIONS, QUEST\_ENGINE
6. **Transaction Logger** → SHOP\_CHECK, inventory events