

Intro:

Environment Choice:

This project uses a web-based interface designed for desktop browsers. The web environment allows quick access for both guest and registered users.

Target Users:

Casual players who want to enjoy quick runs as guests and registered players who track scores, achievements, and seasons.

Design Explanation:

Each wireframe represents one key feature mapped to the project database (Player, Session, Season, Skin, Achievement). Yellow comments describe the purpose and expected function of every component—what it does, how users interact with it, and what they can expect as output

Slime Runner

Season 1

Play as Guest

Sign In

Register

This is the first entry screen that identifies the user type and displays the currently active season.

The title "Slime Runner" and season label ("Season 1") show information from the Season table (name, start_date, end_date, is_active = TRUE).

It presents three navigation options for players:

- Play as Guest — creates a temporary player record with account_type = GUEST in the Player table and links directly to 04_Home.
- Sign In — opens the login screen (02_SignIn) for registered users to authenticate using their credentials.
- Register — opens the registration screen (03_Register) for new users to create an account.

Purpose: to separate guest and registered user flows, and to mark the start of a new Session once the player proceeds to gameplay from Home.

Expected behavior: players can enter immediately as guests or authenticate before accessing the main menu.

Offline Slime Runner
Version 1.0

Sign In



The form consists of four rounded rectangular input fields stacked vertically. The top field is labeled "Email Address", the second "Password", the third "Sign In" (which is slightly larger and has a double border), and the bottom field is labeled "Continue as Guest".

Email Address

Password

Sign In

Continue as Guest

[Don't have an account? Register](#)

This page allows registered players to log in using their email and password or continue as a guest.

The input fields correspond to attributes in the Player table:
email, password, and account_type = REGISTERED.

- Sign In button — validates credentials and, on success, redirects to 04_Home, where the player can view the dashboard and start a new Session.
- Continue as Guest — skips authentication, creates a temporary guest player (account_type = GUEST), and links to 04_Home as well.
- “Don’t have an account? Register →” link — navigates to 03_Register, where new users can sign up.

Purpose: to authenticate or bypass login based on player type while maintaining data integrity in the Player table.

Expected behavior: correct login sends the player to Home; wrong input would normally trigger validation (not implemented in the mockup).

Useername

Email

Password

Create Account

Back to Sign In

This screen enables new players to register an account by entering a Username, Email, and Password, all of which map directly to fields in the Player table (username, email, password).

On pressing Create Account, a new player record is created with account_type = REGISTERED, created_at set to the current date/time, and last_login_at initialized.

Navigation links:

- Create Account — redirects to 04_Home after successful registration, where the player can start playing.
- Back to Sign In — navigates back to 02_SignIn if the user already has an account.

Purpose: to establish a permanent user record for tracking sessions, achievements, and skins.

Expected behavior: data is stored to the Player table, allowing future sessions and achievements to link back to this account.

Home

Welcome back, username!

Active Season: Season 1

Start Run Skin

Achievements Leaderboard

Run History Setting

Last Run

Score

Distance

Top Speed

Crash Type

This is the main hub of the game after login or guest entry. It summarizes the player's recent activity and provides access to all game features.

The header greets the player using their Player.username, and the active season banner shows data from the Season table (name, start_date, end_date, is_active = TRUE).

The Last Run panel displays results from the most recent Session record (fields: score, distance_m, top_speed, crash_type).

Main navigation buttons link to other screens:

- Start Run → 10_GamePlay — begins a new gameplay session, creating a new Session record.
- Skins → 05_Skins — opens the skin selector to choose a skin for the next session.
- Achievements → 06_Achievements — shows unlocked and locked achievements.
- Leaderboard → 07_Leaderboard — opens the ranking list by season.
- Run History → 08_History — displays all previous sessions.
- Settings → 13_Settings — allows editing user preferences.

Purpose: to centralize navigation and summarize performance. Expected behavior: clicking a button navigates to the respective screen while preserving player session context.

Label: "Current Active Skin: Desert Runner"

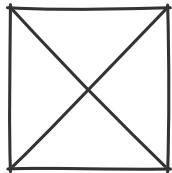
Confirm

Back

Skins

Owned

Locked



Skin Name: Desert Runner

Rarity: Rare

Use Button

This screen displays all available character skins and indicates ownership status.

It uses data from Skin (name, rarity, is_default) and PlayerSkin (player_id, skin_id, acquired_at, source).

Owned skins appear as selectable cards with a "Use" button; locked skins show a lock icon and are disabled.

The Confirm button saves the selected skin as active for the next session (updates Session.skin_id).

Navigation links:

- Confirm → 04_Home — confirms choice and returns to main dashboard.
- Back → 04_Home — returns without saving changes.

Purpose: to let players visually choose and equip a cosmetic skin while ensuring database integrity (Rule R4: player must own skin before use).

Expected behavior: selecting a skin highlights it as the active one for the next run.

Label: "Current Active Skin: Desert Runner"

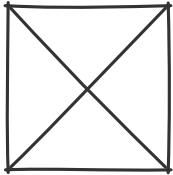
Confirm

Back

Skins

Owned

Locked



Skin Name: Desert Runner

Rarity: Rare

Locked

This screen displays all available character skins and indicates ownership status.

It uses data from Skin (name, rarity, is_default) and PlayerSkin (player_id, skin_id, acquired_at, source).

Owned skins appear as selectable cards with a "Use" button; locked skins show a lock icon and are disabled.

The Confirm button saves the selected skin as active for the next session (updates Session.skin_id).

Navigation links:

- Confirm → 04_Home — confirms choice and returns to main dashboard.
- Back → 04_Home — returns without saving changes.

Purpose: to let players visually choose and equip a cosmetic skin while ensuring database integrity (Rule R4: player must own skin before use).

Expected behavior: selecting a skin highlights it as the active one for the next run.

Achievements

Unlocked

Locked

Allows user to switch between unlocked and locked

eg.

Marathon Runner — Run 1000m without collision

Unlocked at 2025-10-10

Session →

Survivor — Play for 5 minutes without crashing

Locked

[View Session](#)

Shows all game achievements defined in the Achievement table (name, description) and their unlock status for the current player via the PlayerAchievement table (unlocked_at, session_id).

The screen uses two tabs or toggle buttons: Unlocked and Locked.

Each unlocked achievement displays its name, description, and unlock date. A "View Session" link opens the detailed session that earned it.

Navigation links:

- View Session → 09_SessionDetail — opens detailed view of that specific session.
- Back to Home → 04_Home — returns to the main dashboard.

Purpose: to motivate players and visualize progress through milestones tied to gameplay data.

Expected behavior: achievements dynamically update after each session; locked ones show only conditions, unlocked ones include timestamps and links.

Unlocked achievements are shown with their date and session link.

Achievements

Unlocked

Locked

Allows user to switch between unlocked and locked

eg.

Marathon Runner — Run 1000m without collision

Unlocked at 2025-10-10

Session →

Survivor — Play for 5 minutes without crashing

Locked

[View Session](#)

Shows all game achievements defined in the Achievement table (name, description) and their unlock status for the current player via the PlayerAchievement table (unlocked_at, session_id).

The screen uses two tabs or toggle buttons: Unlocked and Locked.

Each unlocked achievement displays its name, description, and unlock date. A "View Session" link opens the detailed session that earned it.

Navigation links:

- View Session → 09_SessionDetail — opens detailed view of that specific session.
- Back to Home → 04_Home — returns to the main dashboard.

Purpose: to motivate players and visualize progress through milestones tied to gameplay data.

Expected behavior: achievements dynamically update after each session; locked ones show only conditions, unlocked ones include timestamps and links.

[Back to Home](#)

Select Season

Season 1

▼

Season 2

[View My Session](#)

Leaderboard

Season 1

e.g.

1. SlimeMaster — 4850 pts — 2025-10-10
2. DinoKing — 4600 pts — 2025-10-09
3. Guest-1024 — 4300 pts — 2025-10-08
- .
- .
- .
- .

Your Rank: #12 | Score: 2500

Displays a ranked list of top players within the selected season, combining data from Session, Player, and Season tables.

The Season Dropdown pulls all Season.name values and allows switching between active and past seasons.

The Leaderboard Table lists fields:

rank, Player.username, Session.score, and Session.ended_at.

The My Rank area highlights the current player's rank and score for quick reference.
Navigation links:

- View My Session → 08_History — opens the player's run history page.
- Back to Home → 04_Home — returns to dashboard.

Purpose: to show fair seasonal competition and reflect cumulative performance.

Expected behavior: table updates when a new season is selected; player's position is visually highlighted.

[Back to Home](#)[View Selected Session](#)

Run History

All session played by you

Season

Crash Type

Date Range

input To [input](#)"/>

- 2025-10-11 | Score 2350 | Distance 785m | COLLIDE [View Details](#)
- 2025-10-10 | Score 1800 | QUIT [View Details](#)
- 2025-10-09 | Score 2500 | TIMEOUT [View Details](#)

Total Sessions: 28

Average Score: 2100

Longest Distance: 980m

Displays a list of all sessions played by the current player, using data from the Session table (session_id, started_at, score, distance_m, top_speed, crash_type, device_type, season_id).

The top Filter Bar allows narrowing down results by:

- Season (Season.name, Season.season_id)
- Crash Type (COLLIDE, QUIT, TIMEOUT)
- Date Range (started_at, ended_at)

The main Table / List shows each session's date, score, distance, speed, and crash type.

Each entry includes a "View Details" button that links to 09_SessionDetail, where the full event timeline is shown.

The bottom section displays summary statistics: total sessions, average score, and longest distance — all aggregated from Session data.

Navigation links:

- View Details → 09_SessionDetail — open detailed record.
- Back to Home → 04_Home — return to dashboard.

Purpose: to help players review and compare their past performance.

Expected behavior: users can filter or browse past runs and open individual details for deeper analysis.

Session Detail

Session ID:

Score

Distance

Top Speedd

Crash Type

Skin

Device

00:00 — START

00:350 — Obstacle GROUND: speed 20px/s (cleared: yes)

00:670 — Input: JUMP

01:020 — Obstacle AIR: speed 22px/s (cleared: no; crash: yes)

01:025 — Session End (COLLIDE)

Provides an in-depth view of one specific gameplay session, combining data from Session, ObstacleSpawn, InputEvent, and ObstacleType tables. The Session Summary Panel at the top shows data from the Session record: score, distance_m, top_speed, crash_type, skin_id, and device_type.

The Timeline List below merges events in chronological order by t_offset_ms:

- ObstacleSpawn entries include obstacle type, speed, and whether cleared.
- InputEvent entries show player actions (JUMP, DUCK, PAUSE) and timing.

This timeline helps identify when obstacles appeared, how the player reacted, and what caused the crash.

A small Slider / Replay Progress bar visually represents the session duration (Session.duration_ms).

Navigation links:

- Back to History → 08_History — return to the Run History list.

Purpose: to visualize the gameplay sequence for debugging or learning.

Expected behavior: users can review key actions and events that occurred during a specific run.

Score

Pause / Resume

Distance

Game Over

Speed

Time

[GAME AREA — Character + Obstacles]

Jump

Duck

Pause

The live gameplay interface where players control the character to jump and avoid obstacles.

The HUD (Heads-Up Display) at the top shows real-time session data from the Session table:

score, distance_m, top_speed, and gameplay time.

The Game Area represents the side-scrolling environment where ObstacleSpawn events occur and player actions are logged as InputEvent records.

The bottom Control Buttons (Jump, Duck, Pause) visualize available actions — each corresponds to an InputEvent (action, source=KEYBOARD).

The Pause Button in the HUD links to 11_Pause, opening the overlay menu.

When the game ends, the player is automatically redirected to 12_GameOver, with final data saved in Session (ended_at, crash_type, score).

Navigation links:

- Pause → 11_Pause — open pause overlay.
- Game Over trigger → 12_GameOver — show results summary.

Purpose: to present the core gameplay interaction and record user performance data.

Expected behavior: players see live stats update during gameplay; pause and game over transitions occur smoothly.

Game Paused

Resume

Restart

Quit

[PAUSED — Overlay Layer]

An overlay that appears on top of the Game View to indicate the game is paused.

The dark background covers the GamePlay screen, making the pause menu the focus.

Three buttons provide clear options:

- Resume — closes overlay and returns to 10_GamePlay; gameplay continues.
- Restart — restarts gameplay by creating a new Session record; also redirects to 10_GamePlay.
- Quit — ends the session with `crash_type = QUIT`, sets `ended_at`, and navigates to 12_GameOver.

Data connection: updates the Session record if the player quits mid-run.

Navigation links:

- Resume → 10_GamePlay
- Restart → 10_GamePlay (new session)
- Quit → 12_GameOver

Purpose: to pause gameplay safely without losing context or progress.

Expected behavior: overlay appears semi-transparent; player resumes or exits seamlessly.

[Play Again](#)[Review Run](#)[Back to Home](#)

Game Over

Session Complete — See your results below

Score
Distance
Top Speed
Crash Type
Skin Used
Device

New Achievements Unlocked

Marathon Runner — Run 1000m without collision
Unlocked at: 2025-10-11

Displays results for the most recently completed session.

The Summary Panel shows Session fields: score, distance_m, top_speed, crash_type, skin_id, and device_type. Below the summary, the Achievements Section lists any new achievements unlocked during this session, retrieved from PlayerAchievement (unlocked_at, achievement_id, session_id) joined with Achievement (name, description). Buttons at the bottom allow the player to continue:

- Play Again → 10_GamePlay — start a new session.
- Review Run → 09_SessionDetail — open detailed event timeline.
- Back to Home → 04_Home — return to main dashboard.

Purpose: to conclude the game loop, show results, and encourage replay.

Expected behavior: summary data displays immediately after a session ends; achievements update dynamically.

[Save Changes](#)[Log Out](#)

Settings

Username Emails Account Type Member Since

Default Skin:

[Classic](#) [Desert Runner](#) [Night Mode](#)[Set as Default](#)

The Settings page allows players to manage their account information and game preferences. It retrieves and updates data from the Player table (username, email, account_type, created_at) and connects with Session and PlayerSkin for certain preference settings.

The page is divided into three main sections:

- Account Information — displays editable fields for username and email, plus read-only fields for account_type (REGISTERED / GUEST) and member_since (Player.created_at). Updates are saved to the Player table.

- Gameplay Preferences — contains checkboxes and toggles for options such as:

- Enable Sound (visual-only UI toggle, affects user experience)

- Auto Sync Offline Sessions — corresponds to Session.is_offline, controlling whether offline data is uploaded when reconnected.

- Dark Mode and Show FPS — optional UI preferences (local display only).