User Story Mapping
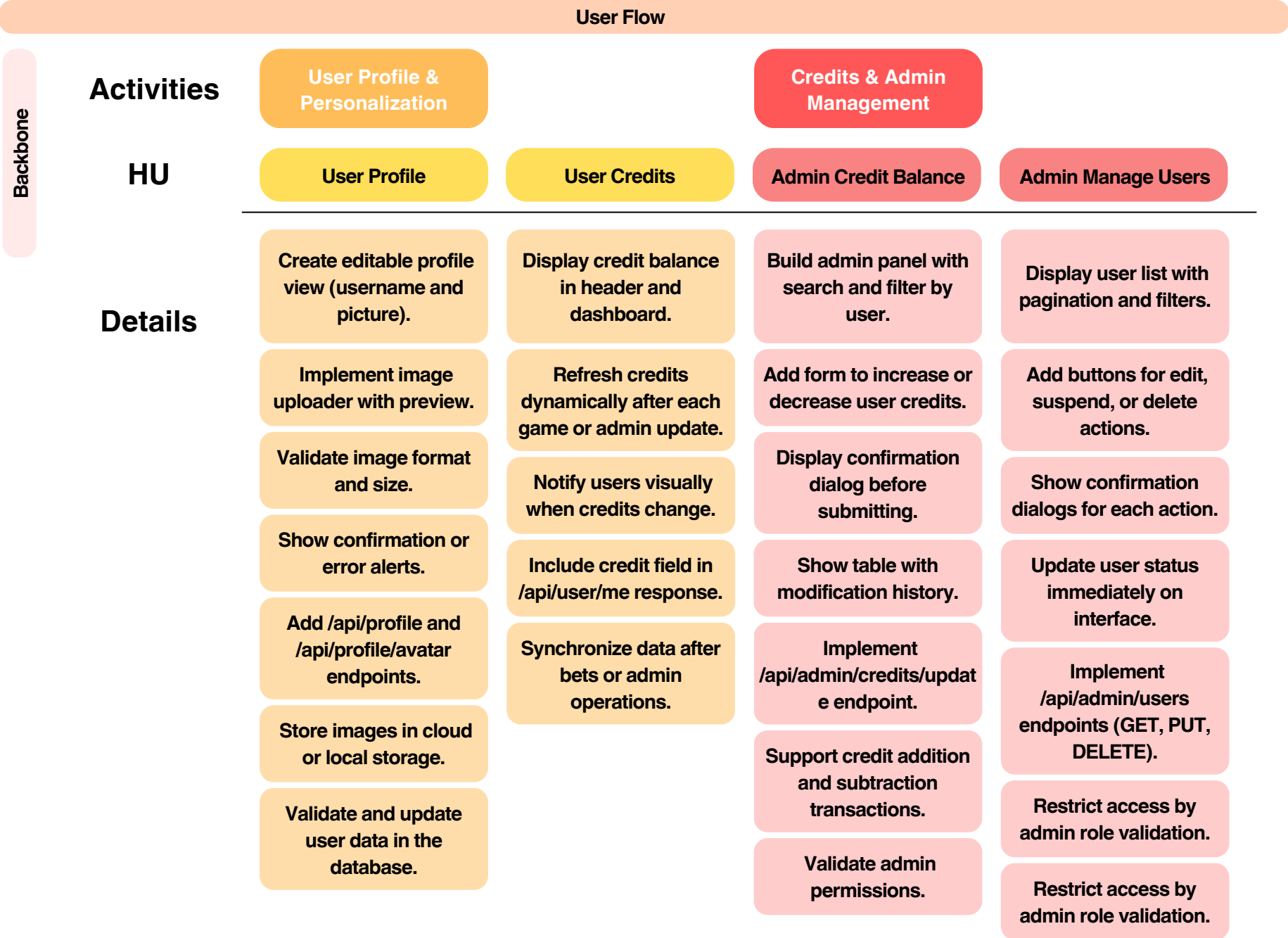
User Flow			
Activities	**Authentication & Access**		
HU	**User Sign Up**	**User Log In**	**Password Recovery**	**Logout**

(Backbone — left side label)

Details	User Sign Up	User Log In	Password Recovery	Logout
	Create registration form with fields (username, email, password)	Create login form with fields (username and password)	Add "Forgot Password" link in login screen.	Add logout button accessible from all screens
	Validate input on client and server	Handle incorrect credentials and loading feedback	Collect user email and show confirmation message.	Clear authentication tokens
	Display success/error messages	Store and manage authentication tokens in session or local storage.	Display reset password form when link is opened.	Redirect to the login page
	Implement /api/register endpoint	Create /api/login endpoint	Implement /api/forgot-password and /api/reset-password routes.	Invalidate tokens (blacklist or delete session)
	Hash passwords	Verify credentials securely against hashed passwords	Generate and validate secure recovery tokens.	Clear active sessions in server memory or database.
	Validate email uniqueness and password strength	Generate and return JWT or session token.	Send recovery link by email.	Register logout events
	Log registration events in the database	Handle login lockouts after multiple failed attempts.	Update password hash after successful reset.	

User Story Mapping

User Flow				
Backbone				
Activities	User Profile & Personalization		Credits & Admin Management	
HU	User Profile	User Credits	Admin Credit Balance	Admin Manage Users
Details	Create editable profile view (username and picture).	Display credit balance in header and dashboard.	Build admin panel with search and filter by user.	Display user list with pagination and filters.
	Implement image uploader with preview.	Refresh credits dynamically after each game or admin update.	Add form to increase or decrease user credits.	Add buttons for edit, suspend, or delete actions.
	Validate image format and size.	Notify users visually when credits change.	Display confirmation dialog before submitting.	Show confirmation dialogs for each action.
	Show confirmation or error alerts.	Include credit field in /api/user/me response.	Show table with modification history.	Update user status immediately on interface.
	Add /api/profile and /api/profile/avatar endpoints.	Synchronize data after bets or admin operations.	Implement /api/admin/credits/update endpoint.	Implement /api/admin/users endpoints (GET, PUT, DELETE).
	Store images in cloud or local storage.		Support credit addition and subtraction transactions.	Restrict access by admin role validation.
	Validate and update user data in the database.		Validate admin permissions.	Restrict access by admin role validation.

User Story Mapping

	User Flow			
Backbone				
Activities	**Roulette Game Flow**		**Slot Machine Game Flow**	
HU	**Betting on Roulette**	**Roulette Spin and Results**	**Start Slot Game**	**Prizes and Winning Combinations in Slots**
Details	Create interactive roulette board with chips.	Animate roulette wheel with smooth transitions.	Display slot reels and "Spin" button.	Add "Pay Table" button in slot interface.
	Enable selection of numbers, colors, or ranges.	Show the winning number and color on screen.	Show spinning animation and disable buttons during play.	Display winning combinations and payout values.
	Disable "Spin" until bet is valid.	Display pop-up with result and updated balance.	Present visual results and winning highlights.	Highlight matching symbols after spin result.
	Deduct credits visually after confirmation.	Generate random outcome securely	Implement /api/games/slots/spin endpoint.	Serve /api/games/slots/paytable route.
	Implement /api/games/roulette/bet route.	Calculate winnings based on bet rules.	Generate random reel outcomes.	Define payout ratios for symbol patterns.
	Validate user balance and bet type.	Update balance and mark bet as settled.	Calculate and return payout.	Retrieve data dynamically based on bet amount.
	Record bet in bets table.	Broadcast result via WebSocket	Update user balance accordingly.	
	Deduct credits within a transaction.			

User Story Mapping

User Flow

Activities

| User Feedback & Tracking | | Administrative Tools & Analytics |

HU

| Betting History | Game Rules / Help Section | System Statistics Dashboard |

Details

Betting History	Game Rules / Help Section	System Statistics Dashboard
Create page to view past bets with filters by date and game.	Design help page with clear layout and navigation.	Build dashboard with charts (users, bets, payouts, credits).
Display columns for amount, result, and date.	Include detailed instructions and visual examples.	Implement filters by date and game type.
Implement /api/bets/history endpoint.	Link help section from each game interface.	Auto-refresh statistics periodically.
Return paginated list of bets with outcomes.	Store help text and rules as static HTML	Create /api/admin/stats route.
Filter by game type and date range.	Serve data via /api/help endpoint.	Aggregate totals from database tables.
		Secure route using admin authentication