

# Wikisportipedia by HeistLEET

## TARGET SHIP DATE: 2025-11-30

Wiki About: Sports

entries

int	sport_id	PK
string	name	
string	description	
int	team_size	
date	created_on	

account

int	account_id	PK
string	username	
string	email	
string	username	
date	created_on	
string	first_name	
string	last_name	

history

int	sport_id	
int	account_id	
date	edited_on	

Components:

- SQLite file containing the table **accounts** to store users of the wiki and their appropriate data, the table **entries** to contain entry data, and the **history** table to contain the records of who has changed each wiki page and when.

- Routes:
  - /sign-in, verifies values against **accounts** table, redirects to / with cookies when done, uses **fetch\_creds**
  - /register, uses **fetch\_creds()** to verify account does not exist, adds values to **accounts** with **set\_creds()**, redirects to sign-in
  - /, Ninja2 wikipedia-style search bar that uses **fetch\_page** to search for pages and redirect
  - /account, if not signed in redirects to sign-in, if the cookies are present displays content of the **accounts** record for the user and queries **entries** for where they have worked, uses **fetch\_creds()** as well as **fetch\_history()**. The results of **fetch\_history** are fed into the links using iteration.
  - /<page>, uses **fetch\_page(page)**, checks the value of page against the titles in the **entries** base, if present display a rendered template with the content plugged into the page to form a complete page, if not redirect back to /
  - /<page>/edit
    - If the user is signed in and the page is valid they are presented with a textbox to edit the content of the page in, when submitted the **entries** record is updated, uses **fetch\_creds**
  - /new, provides a form for logged in users to submit a new page to the site, uses **fetch\_page** to first verify if the page exists, if not then uses **set\_page** on the values
- Helper Functions:
  - **fetch\_page(page)**, returns a dictionary containing the values needed to populate a page from the sql, returns false if the page does not exist
  - **set\_page(page)**, sets values into **entries** table, adds record of the change to the **history** table
  - **create\_page(...)**, enters a new page into the **entries** table, also logs in **history**
  - **fetch\_creds(username)**, returns dictionary of user information, if the user isn't signed in then this is bypassed
  - **set\_creds(...)**, sets values to **accounts** table
  - **fetch\_history(username)**, returns all **distinct** page ids the user has edited

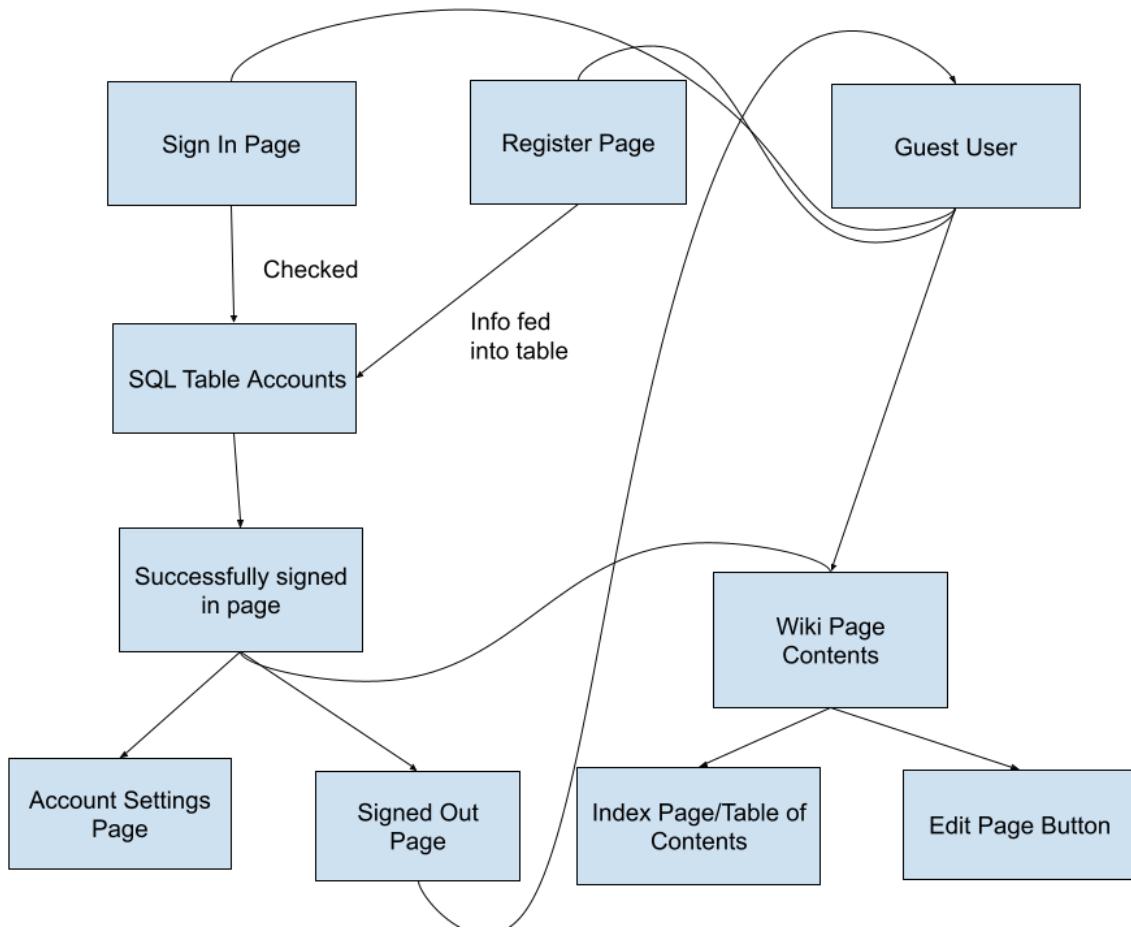
#### Site Map:

- /sign-in
  - Cookie functionality as we created in class for assignment #16
  - Account credentials should be checked against the SQL table **accounts** from this route
  - Link to /register for new users
  - If account credentials are verified, we redirect to / with cookies set
  - If account credentials are invalid, we reload with an error message
- /register
  - Again, functionality as created in prior assignment
  - Credentials fed into the **accounts** table, registering their account details (as described in **accounts** schema)
  - Redirects to /sign-in upon successful registration
  - Validation in the accounts table (preventing duplicate accounts)

- /guest
  - Represents browsing mode for users without an **account** or cookies
  - Basically this is just a signed out variant of /
  - Allows read-only access to all public wiki pages (/page/<title>)
  - Cannot access /new, /edit, or /account
  - Redirected to /sign-in if they attempt to go into any restricted domains
  - Links visible: /sign-in, /register, /page/<title>
- /
  - Home/Index page
  - Two variants depending on the sign-in state:
    - Signed in: Greets user with (“Welcome, NAME”), displays Table of Contents of pages (**entries**) + link to /account
    - Signed out: Shows “Sign In” and “Register” buttons
  - Contains search bar at the top that queries entries using the **fetch\_page()** function
  - Links to /page/<title> for each entry
  - Links to /new for creating a page (which is only visible to logged in users)
- /account
  - Displays profile information (Username and Gmail)
  - Lists all pages this user has created or edited (queried from **entries**)
  - Each listed page title links to /page/<title>
  - Button to /sign-out
  - Link to account/settings page for username/gmail/password changes.
- /account/settings
  - Accessible to signed-in users only
  - Allows users to update username, password, or Gmail
  - Updates the **accounts** table fields
  - Redirects to /account after save
- /new
  - Form for logged-in users to submit new wiki pages
  - Checks via **fetch\_page()** if page already exists
  - If not: saves via **set\_page()** into **entries** table
  - Redirects to /page/<title> after creation
- /page/<title>
  - Displays page contents (title, text, author, date) from **entries** table
  - If the user is logged in and the page is theirs they are shown an “Edit Page” button that redirects them to /page/<title>/edit
  - Guests can only view the page
  - If page doesn’t exist then redirect to /
- /page/<title>/edit
  - Displays a textbox pre-filled with the page’s current content
  - Allows logged in users to edit and submit updates
  - Updates the **entries** table upon submission

- Redirects to `/page/<title>` after a successful update
- `/sign-out`
  - Clears user cookies/session
  - Redirects to `/`

Program Components:



## Task tracker

👤 Assignee	Title	Date	Status
rohans33@nycs...	A list of program components with role of each specified. (e.g., a car engine is comprised of various components: carburetor, alternator, radiator, spark plugs, etc. Each must perform its role for the engine to do its overall job.)	Oct 23, 2025	Completed ▾
Michelle Chen	Explanation of how each component relates to the others. <ul style="list-style-type: none"><li>Component map visualizing relationships between components.</li></ul>	Oct 23, 2025	Completed ▾
Aoanul Hoque	Database Organization (tables? Relationships b/t tables? etc.)	Oct 23, 2025	Completed ▾
Haowen Xiao	Site map for front end <ul style="list-style-type: none"><li>Represent each page you envision for your site.</li><li>Show linkages conveying all possible pathways for a user traversing site.</li></ul>	Oct 23, 2025	Not started ▾
👤 Person	A breakdown of the different tasks required to complete this project <ul style="list-style-type: none"><li>Include assignments of each task to each group member</li></ul>	Oct 23, 2025	In progress ▾

## Task tracker

👤 Assignee	>Title	📅 Date	>Status
👤 Person	Append this line to your heading: <b>TARGET SHIP DATE: {yyyy-mm-dd}</b>	Oct 23, 2025	Completed ▾
haowenx2@nyc...	Amalgamate these components into a single PDF, store in designated location.	Oct 23, 2025	Not started ▾