

# Project Info

**Team Name/Project Title:** ThriveToday

**This Weeks Leader:** Henry Fan

**Members:** Alan Luu, Jeremy Esch, Henry Fan (Lead), Taylor Trinidad

**GitHub:** <https://github.com/Jesch101/ThriveToday>

## Job Assignments:

Jeremy: Frontend/Project Setup

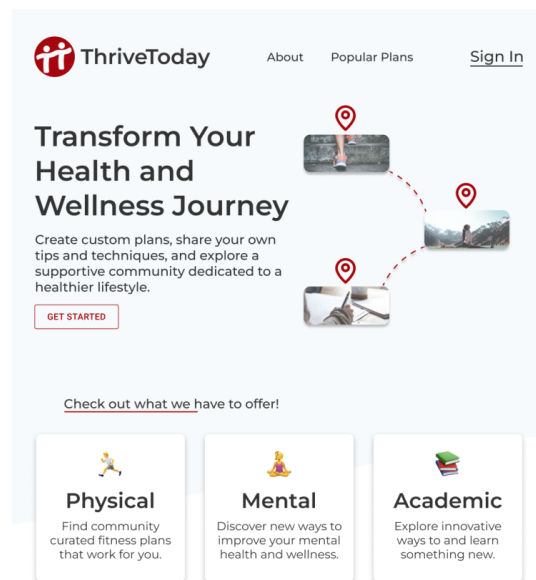
Henry: Backend/ Marketing

Taylor: Backend/UI Design

Alan: Backend/Database

## Application Description

Our application is for learners who realize a thoughtful approach to living a healthy & rich life requires habits that support sustained learning and growth. Our application will reduce the psychological and cognitive load to onboard users with life skills for success in education and beyond. We focus our efforts in providing value for college students because we're students.



## Features List as of 3/2/23

- Plans - Physical Mental Academic
- Like and Unlike buttons
- Navigation Bar
- Sign in / Sign Up

## Functional specifications

### General play description and outcomes

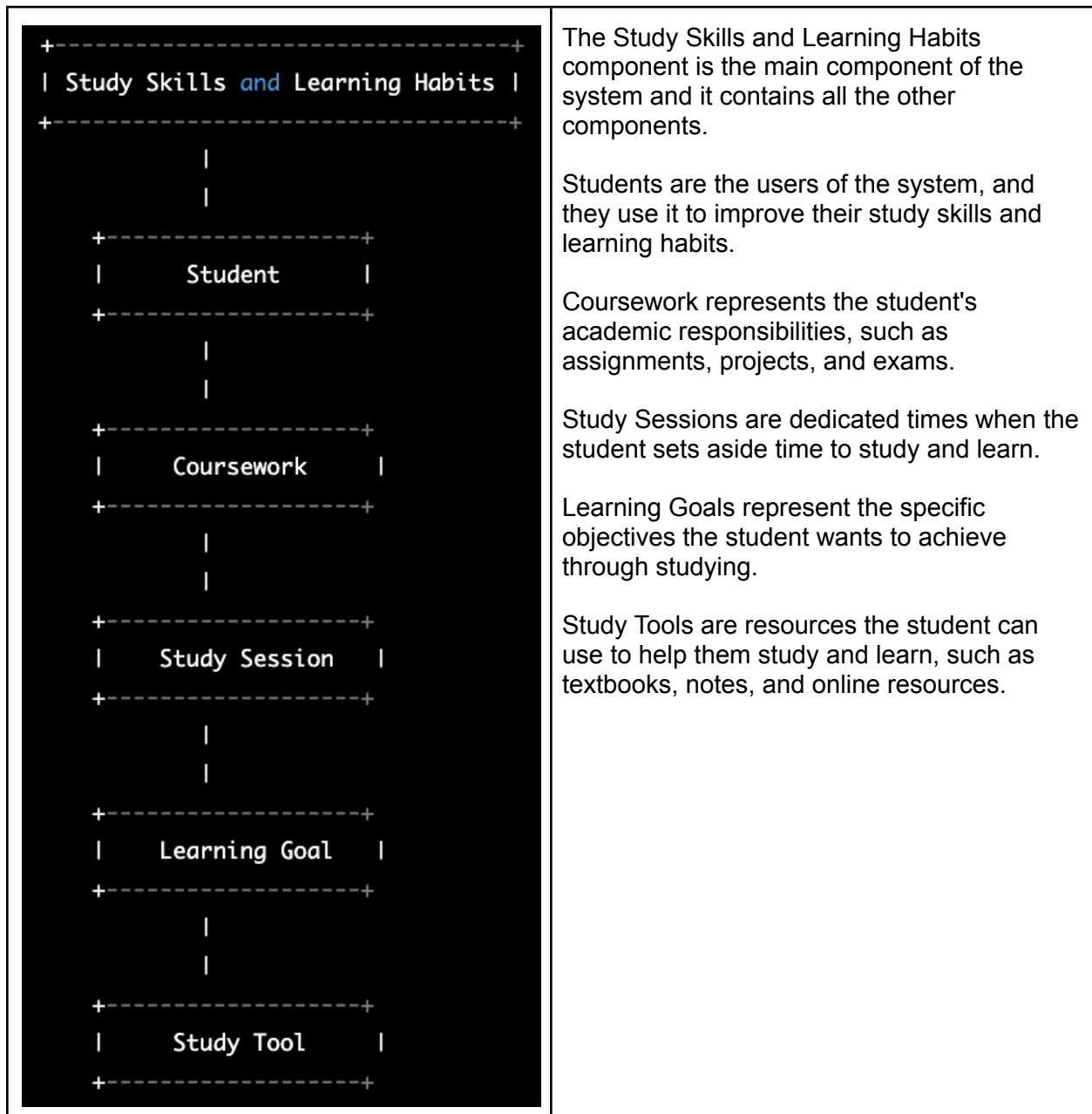
- Users can sign up / sign in to their personal account on our website
- Users will be able to customize plan templates
- Users can post their plans and sub-plans
- Users will be able to like / unlike posts
- Users can filter and sort through varying topics

### Use Cases for each feature

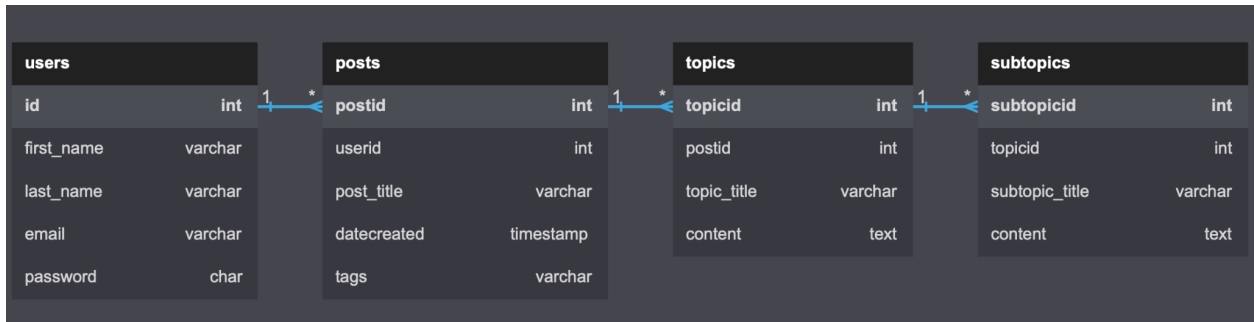
- Plans:
  - **Physical:** Users can access workout plans that cater to different fitness levels and goals, such as weight loss, muscle building, or strength training.
    - Users will be incentivized to provide feedback on community member's plans
  - **Mental:** Users can access plans that focus on improving mental health, such as meditation or stress management techniques.
    - Users will be encouraged to be transparent about their favorite mental health practices
  - **Educational:** Users can access educational plans that help them develop new skills, such as learning a new language or mastering a particular software tool.
    - Users will be guided to Plan Act and Reflect on what how why they are learning
  - **Templates:** Users can access templates for different types of plans, such as workout plans or meal plans, which they can customize according to their needs.
  - **Recommended articles:** Users can access articles related to the plans they are interested in, such as nutrition tips for a workout plan or productivity hacks for an educational plan.
- Like button:
  - Users can use the like button to find the most popular plans, as plans with more likes will be displayed at the top of the list.
  - The like button can also be used to track the popularity of a particular plan over time, as users can see how many likes it has received since it was first posted.
- Navbar:
  - The navbar provides users with easy access to different sections of the website, such as the plans section, the articles section, or the user account section.
  - Users can use the navbar to search for specific plans or articles, or to filter plans by category.
- Sign Up / Sign In:

- Users can create a personalized account, which allows them to save their favorite plans and articles, as well as create their own plans to share with other users.
- Users can also use their account to track their progress with different plans, such as tracking their workout or meditation sessions.

### Sample UML Diagram for Academic Plan



## UML notations, diagrams, etc.



```
CREATE TABLE users(  
    userid SERIAL PRIMARY KEY,  
    firstname VARCHAR(255) NOT NULL,  
    lastname VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    username VARCHAR(255) NOT NULL,  
    password CHAR(60) NOT NULL  
);  
  
CREATE TABLE posts(  
    postid SERIAL PRIMARY KEY,  
    userid INT NOT NULL,  
    posttitle VARCHAR(255) NOT NULL,  
    datecreated TIMESTAMP NOT NULL,  
    tags VARCHAR(255)  
);  
  
CREATE TABLE topics(  
    topicid SERIAL PRIMARY KEY,  
    postid INTEGER NOT NULL,  
    topictitle VARCHAR(255) NOT NULL,  
    content TEXT NOT NULL  
);  
  
CREATE TABLE subtopics(  
    subtopicid SERIAL PRIMARY KEY,  
    topicid INTEGER NOT NULL,  
    subtopictitle VARCHAR(255) NOT NULL,  
    content TEXT NOT NULL  
);
```

## Website Sign Up and Sign In Pages

The image displays two side-by-side wireframes of the ThriveToday website's authentication pages. Both pages share a common header with the ThriveToday logo, 'About' and 'Popular Plans' links, and a 'Sign In' link. The left wireframe is the 'Sign up' page, featuring a red lock icon, a 'Sign up' title, and input fields for 'First Name\*', 'Last Name\*', 'Email Address\*', and 'Password\*'. It includes a red 'Sign Up' button and a link for 'Already have an account? Sign In'. The right wireframe is the 'Sign in' page, featuring a red lock icon, a 'Sign in' title, and input fields for 'Email Address\*' and 'Password\*'. It includes a 'Remember me' checkbox, a red 'Sign In' button, and a link for 'Don't have an account? Sign Up'.

### Testing plans (for each feature)

- Plans
  - Create a plan and make sure all user entries are transferred over to the database and are able to be retrieved
- Like Button
  - Use created test plan and make sure the like button increments the number of likes to the plan by 1 as long as the user has not already liked the plan
  - Test 'un-liking' the plan and remove it from the users likes as well as decreasing the number of likes by 1
- Navbar
  - Create basic navbar and ensure it maps to all pages correctly before adding styling
- Login/Sign Up
  - Create test accounts and make sure both processes work correctly

10 test users:

```
postgres=> select * from users;
userid | firstname | lastname | email | username | password
-----+-----+-----+-----+-----+-----
1 | first1 | last1 | first1@example.com | user1 | password
2 | first2 | last2 | first2@example.com | user2 | password
3 | first3 | last3 | first3@example.com | user3 | password
4 | first4 | last4 | first4@example.com | user4 | password
5 | first5 | last5 | first5@example.com | user5 | password
6 | first6 | last6 | first6@example.com | user6 | password
7 | first7 | last7 | first7@example.com | user7 | password
8 | first8 | last8 | first8@example.com | user8 | password
9 | first9 | last9 | first9@example.com | user9 | password
10 | first10 | last10 | first10@example.com | user10 | password
(10 rows)
```

## Milestones

- Milestone 1: researching tech stack - python, html, css, javascript, SQLite, Node JS/express and PERN
- Milestone 2: exploring various technologies and working through their tutorials
- Milestone 3: Finalized UI design, features list, database schema, learn about tech stack
- Milestone 4: Begin coding frontend and backend (Deadline: 3/4)
- **Milestone 5:** Start testing API and set up local hosting (Deadline: 3/18)
- Milestone 6: Connect database to backend and start on responsive frontend design (Deadline: 4/1)
- Milestone 7: Integrate frontend with backend through Axios API request library (Deadline: 4/15)
- Milestone 8: Prototype (Deadline: 4/29)