App Idea:

WishSave AI is a budgeting and savings goal tracker designed specifically for middle and high school students. The app helps users simulate a real-world income, track weekly expenses, and set long-term savings goals (like a new bike or gaming console). It introduces foundational financial literacy concepts in a fun, visual, and interactive format. Students log expenses by week and watch their progress toward a financial goal through dynamic progress bars and projections. Students could also use it as a real simple finances manager instead of just a learning tool.

How is AI/ML playing a role in your app?

AI plays the role of a virtual **savings coach**. Using a local large language model (LLM) via Ollama (OpenAI API would not work) the coach analyzes the user's income, expenses, and goals, then provides personalized advice. Users can ask questions like "What can I cut from my spending?" or "How can I save faster for my goal?" and receive smart, encouraging answers based on their actual financial input. Users can also ask math questions to find out how much more they would need to save to buy a certain item within "x" amount of weeks. This makes budgeting feel conversational and makes details easy to understand for the youth.

What were the considerations when developing the user interface?

The UI was made to be simple

- **Minimal navigation** using a TabView (Home, Expenses, Goal, Coach)
- **Interactive elements** like segmented weekly views, a fillable progress bar, and labeled expenses
- **Real-world simulation** by including job titles, income sliders, taxes, and inputable expense categories
- **Simple Words,** for student understanding
- **Accessibility** with readable fonts, intuitive layout, and minimal text overload

We prioritized helping users “see” their financial journey at a glance, without needing financial jargon or a high learning curve.

Are there similar apps out there? How does yours compare?

Most budgeting tools on the market are either designed for adults managing real bank accounts or for parents supervising teen spending. These platforms tend to prioritize automation, financial complexity, or parental control. WishSave AI takes a different approach by focusing on hands-on learning, simplicity, and student independence. Instead of tracking real money, it uses virtual income and spending to teach budgeting basics in an age-appropriate, low-pressure environment. The app emphasizes goal-setting and progress tracking, helping students understand how everyday decisions impact long-term savings, and big purchases. Most other budgeting apps for younger users are not interactive enough to teach saving habits in context.

- **Mint:** Focuses on syncing real bank accounts and tracking expenses passively. WishSave is manual on purpose, to encourage learning and conscious input.

- **GoHenry & Greenlight:** Apps that help parents manage kids’ money with prepaid cards. WishSave doesn’t require a card or real money — it’s a learning simulation ideal for classroom or personal development. However, this takes the decision making away from the student, meaning they do not develop good habits because they don’t practice them.

Prompts:

1. Create a SwiftUI TabView with four tabs: Home, Expenses, Goal, and Coach. Each tab should have its own view struct and system image label.

2. Build a reusable BudgetModel class using ObservableObject with @Published properties for job title, annual income, expenses per week (array), and savings goal name and amount.

3. In the Home tab, display monthly income before tax and after tax (15% deduction) based on the user’s annual income input. Format the numbers as currency and separate each section

4. In the Expenses tab, use a control to let users switch between Weeks 1–4. Under the picker, include a text field for entering expense title and another for amount, then store the expense in the correct week.

5. Display a list of expenses below the input section, grouped by week.

6. Add a line at the bottom of the Expenses tab that shows the user’s adjusted monthly income (monthly income after tax minus total expenses). Display this value in bold.

7. Create a vertical progress bar in the Goal tab that updates based on how close the user is to reaching their savings goal. Do the math to find the amount of weeks it would take

8. In the Goal tab, show the percentage of goal progress, amount remaining, and an estimate of how many months it will take to reach the goal. These values should update live based on the user’s inputs and spending history.

9. Make a progress bar in the goal tab Add a label underneath that says Progress: ___% with the percentage calculated from saved vs. goal amount.

10. In the Coach tab, create a text editor where the user can type a question, a button labeled 'Ask', and a scrollable text area to show the AI's response.

11. Style the AI response area with padding, a light gray background, and rounded corners to visually separate the answer from the rest of the screen. Make it simple and easier to see using contrast

12. Add a “Suggested Weekly Spending Limit” text at the bottom of the Expenses tab. This should calculate 70% of available weekly income and update whenever new expenses are added. Use a blue font to highlight it.