

# CGPA Calculator and Currency converter

## web application

### Purpose:

Two core utilities: (Cumulative Grade Point Average calculator),(Currency convertor).

Cgpa calculator: allows users to input their grades and credits to calc their cgpa

Currency convertor:enables the user to convert the currency into their desired currency (quick and efficient currency convertor).

### Frontend:

- React.js for user interface (learning)
- React router for navigation between features (have to learn)
- Tailwind css for styling the webpage

### Api:

- exchangerate.host free it seems , no public key is needed , realtime currency exchange

### Testing:

- Make sure both CGPA Calculator and Currency Converter work correctly
- Try edge cases (e.g., empty inputs, invalid entries)

### UI Polish:

- Check layout and responsiveness (especially on mobile)
- Add basic styling if not done already (Tailwind/Bootstrap)

i have already done with 5  
i would like to do 1

yes you are right i was going to do login page and save the details then when the user changes or updates the gpa or cgpa we can keep the track of it

and this login should be optional

i will make the user login if they want to track the progress by adding their gpa and cgpa  
and will suggest them to score good gpa according to their desired cgpa

if the user just want to check their cgpa or gpa for once  
i would leave them to check by skipping the login  
if the user skipped the user details and tracking should not work  
which is basically when the user logged in then the backend should work and should track and suggest stuff

i would like to use MERN  
is it good?

and i have only 6days to do it  
can i be able to complete this task

and do not forget that i am just a beginner and i am learning while completing the task  
and as you know i am new to react  
i am also new to mongo db express and node

is it ok to use these or do u have any suggestions?

You're thinking like a real product builder now — optional login, user-specific tracking, and personalized suggestions? That's a fantastic upgrade!

✓ About MERN: Is it a good choice?

Yes — the MERN stack is a great full-stack option for your project:

- MongoDB – stores user info and GPA/CGPA history.
- Express.js – backend API to handle login, data saving, and suggestions.
- React.js – frontend UI (you already know this part).
- Node.js – server environment for running Express.

Even though you're a beginner, this stack has tons of beginner-friendly tutorials, and since you're already comfortable learning while building, you can definitely do this with good planning.

---

Can you finish in 6 days?

Yes — but only if you follow a clear, focused roadmap and avoid unnecessary complexity. Here's a tight plan:

---

## 6-Day Plan (Beginner-Friendly MERN)

### ✓ Day 1: Setup & Theme Toggle

Add Dark Mode toggle using Tailwind (dark: class).  
Setup MongoDB Atlas, Node.js + Express server.  
Create basic folder structure: /client for React, /server for backend.

### ✓ Day 2: Build Login (Optional)

Create login and signup pages.  
Allow "Continue without login" as a choice.  
Store login state in localStorage.

### ✓ Day 3: Backend Auth

Use JWT for basic token-based login.  
Create Express routes: /signup, /login, /userdata.  
Test it with Postman or simple frontend forms.

### ✓ Day 4: GPA/CGPA Tracking

If logged in, store GPA/CGPA updates to MongoDB.  
Create route: POST /progress, GET /progress.  
Show previous data if logged in.

### ✓ Day 5: Smart Suggestions

Based on CGPA goal, suggest minimum GPA to maintain.  
Example: "You need a 9.1 GPA next semester to hit your goal CGPA of 8.5."  
Add this logic on backend or frontend.

### ✓ Day 6: Polish & Test

Fix bugs, clean UI.  
Add loading indicators, toasts for alerts.  
Host frontend on Netlify/Vercel, backend on Render/Glitch (easy options).