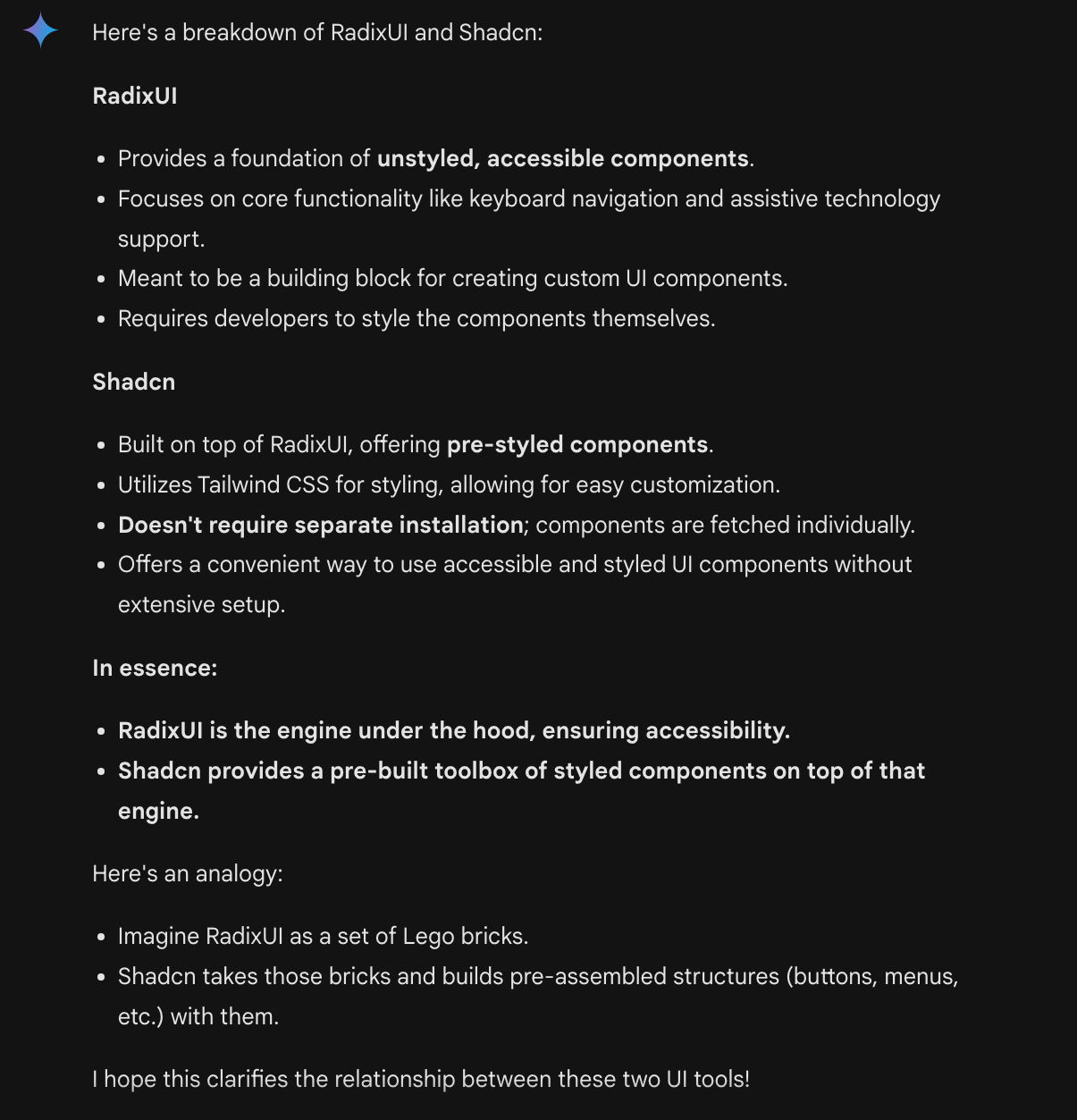
**Week 8**

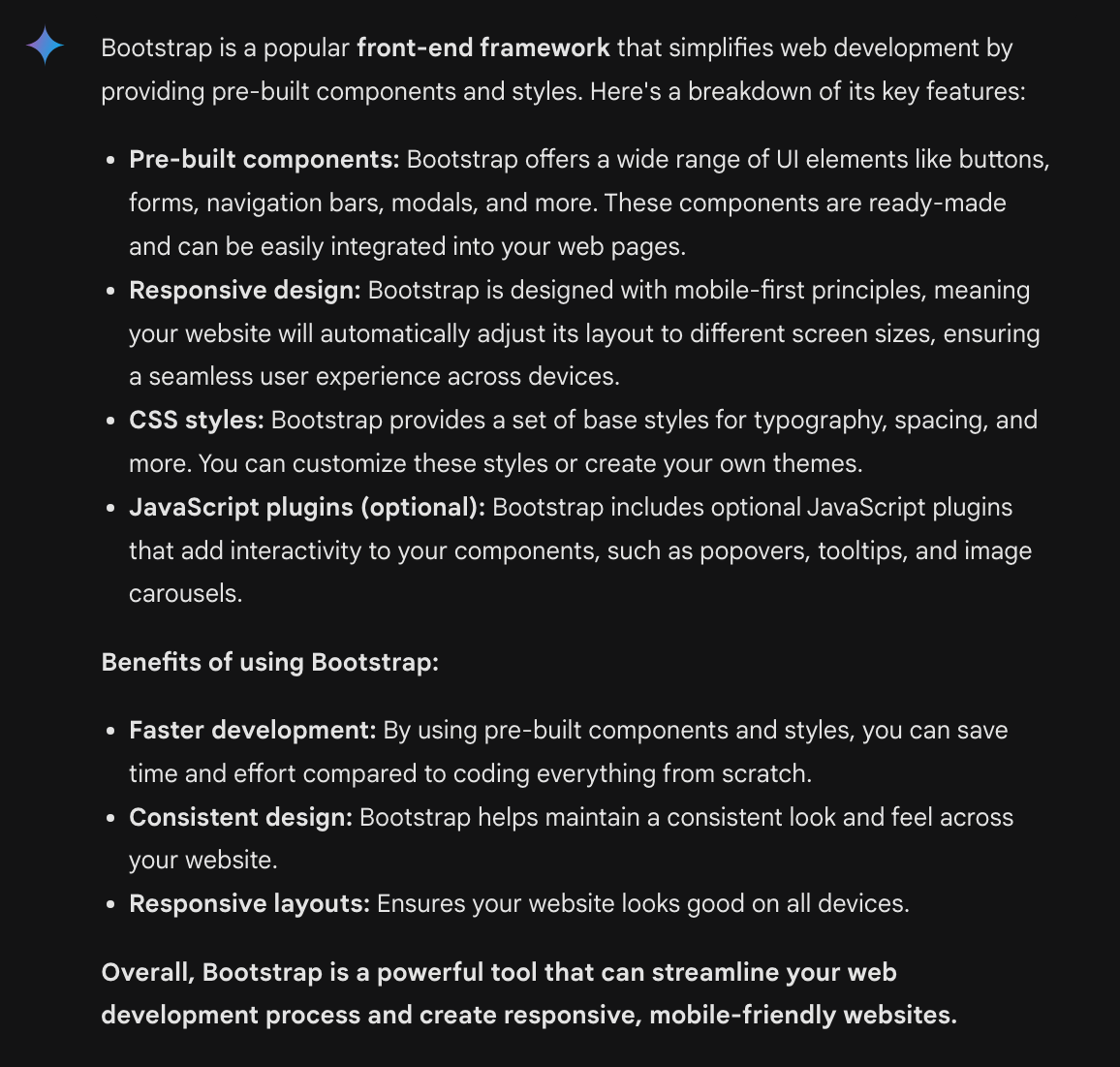
**8.0 Tailwind**

* Flex-start is default.
* We set the default CSS and the CSS to give below the breakpoint.
* The parent div should have grid grid-cols-12 and the child div should have col-span-2.

**8.1 Tailwind**

* Flex lets you position divs right next to each other.
* In react, you have to give style as object.
* We cannot write class because class is a reserved word in JS. So, we write className.
* The objective of grids can be achieved using flex and w-[40%]. But, it’s better to use col-span.
* Breakpoints come under responsiveness.
* Before sm breakpoint, the size is small and so on.
* What this means is that unprefixed utilities (like uppercase) take effect on all screen sizes, while prefixed utilities (like md: uppercase) only take effect at the specified breakpoint and above.
* Use unprefixed utilities to target mobile and override them at larger breakpoints.
* Use Flowbite for sidebar.
* If you want a part of your codebase to be open sourced, you can use storybook.
* It also brings a lot of dependencies and errors.
* We need to write stories for that component so that it appears in the storybook UI.
* Stories are files that need to be written by the UI engineer.
* MUI hinders customization. It gives you components out of the box but they are very hard to customize.
* Overriding it gets very ugly.
* MUI is highly opinionated. Tailwind is not opinionated.
* If you want to quickly get a website running then its good, like in a hackathon or something.
* Eventually, you want to use Tailwind. More specifically Shadcn and RadixUI. You would create a UI library of your own that is customizable in the future.
* Vanilla CSS 🡪 CSS without any frameworks.





**8.2 PayTM Backend**

* [Dailycode (daily-code-web.vercel.app)](https://daily-code-web.vercel.app/tracks/oAjvkeRNZThPMxZf4aX5/QDisg3v6Fo9r08H6NsSd)
* Building a docker image: docker build ./ -t mongodb:4.7-replset
* Running it: docker run --name mongodb-replset22 -p 27017:27017 -d mongodb:4.7-replset
* To restart the container: docker start mongodb-replset22
* This connect to a local server.
* Connect 🡪 Schema 🡪 Model.
* In database, you always store integers and not floats. You should always know where your decimal point lies.
* startSession 🡪 startTransaction 🡪 commitTransaction
* The later transaction would fail if during its run the read object is written by another previous transaction. This is to make data consistent. This would result in an error, so put this in a try catch block.

**8.3 Axios vs Fetch**

* Fetch is provided natively by the browser and node.
* Axios is an external library.
* In fetch, the converting the response to json is an async task.
* If you’re sending a post request, then the first argument is the URL, the second argument is the body, the third argument is the headers. Same is the case for put and delete. If the request can send body then the second argument is body.
* If you’re sending a get request, the first argument is the URL and the second argument needs to be the headers. However, you can send query parameters.
* In GET request, you can’t send body.
* Useful 🡪 https://github.com/axios/axios

**8.4 PayTM Frontend**

* It’s always better to break down components.
* Store the token in localStorage once you have the token.
* When the user logs out do localStorage.removeItem(“token”).
* We can use Link as well as useNavigate.
* Implement the /me endpoint. It just checks whether the token is valid or not, whether the token has expired or not, etc. If the token is valid, it redirects you to the dashboard else it directs you to the signup page.