AeroAspire Intern

Nithin K Y

Day1/Week2:

Scaffold app; setup basic folder structure; create homepage with MUI Typography and AppBar;

1.What files/folders does Vite produce and what is the build / dev flow?

- index.html sits at the root and loads your app.
- package.json handles scripts and dependencies.
- src/ is where your actual React code lives (main.jsx, App.jsx, components, etc.).
- public/ is for static stuff like images.
- Later, when you run npm install, it pulls in a huge node_modules/ folder automatically.

For the flow — in dev mode (npm run dev), Vite spins up a fast server and serves files directly without heavy bundling. Any time you save, only that part of the app reloads instantly (hot reload), so the feedback loop is super quick.
In build mode (npm run build), Vite switches gears: it uses Rollup to bundle everything, minify, and split code so the final app is small and fast. All of that ends up neatly in a dist/ folder, which is what you'd actually deploy.

2. What is bundling / hot reloading. How does Vite help speed up development?

Bundling is like packing all your JavaScript, CSS, and images into one neat box so the browser can load your app faster. Normally, tools do this even while you're still coding, which can feel slow.

Hot reloading means when you change something (like text or color), the browser updates instantly without refreshing the whole page. It's like editing a Word file and seeing the change immediately instead of closing and reopening it every time.

How Vite helps:

Vite skips the heavy bundling during development. Instead, it serves your files directly to the browser and only rebuilds the small part you changed. That's why edits show up almost instantly, and you don't waste time waiting for long reloads.

3. Describe how React components are structured: what parent/child relationship, what props?

React apps are built using components, like building blocks.

One component can be inside another → this makes a parent-child relationship.

Parents can send data to children using something called props.

The child just uses those props, it cannot change them — this keeps data flow clear and one-way.

my git hub link:- https://github.com/Nithin7333/AeroAspire-Intern.git