1. npx create-next-app@latest .
2. npx shadcn-ui@latest init – creates the global.css file with various color variables (Light + dark) But we can only use hsl values using this, (eg. --background: 0 0% 100%;)
3. npm install next-themes (dark-mode - shadcn) : <https://ui.shadcn.com/docs/dark-mode/next>
4. npx shadcn-ui@latest add input button dropdown-menu

* Installing input, button and dropdown-menu from shadcn

**CONVEX DB -** [**https://docs.convex.dev/quickstart/nextjs**](https://docs.convex.dev/quickstart/nextjs)

Create convex provider and wrap the children component inside it, in app/layout.tsx

A screenshot of a computer

Description automatically generated

A black background with white text

Description automatically generated

Authentication: <https://docs.convex.dev/auth/clerk>

To communicate between clerk and Convex, Webhooks are used

A computer screen shot of a workflow

Description automatically generatedA black background with green text

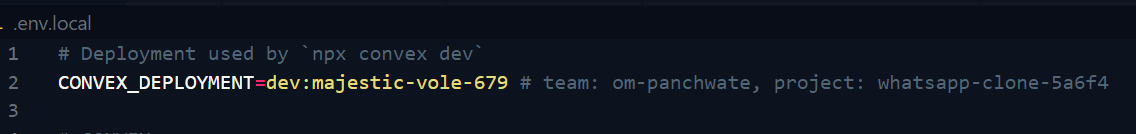
Description automatically generated

1. Clerk detects an event (signup, sign in), Webhook will contain a payload and a POST request will be made to add/update the details in the Convex DB. And a Trigger fulfillment (validate that the request comes from the Clerk)

**NOTE: Webhook can be created in Clerk Dashboard only**

Endpoint URL : <https://docs.convex.dev/functions/http-actions>

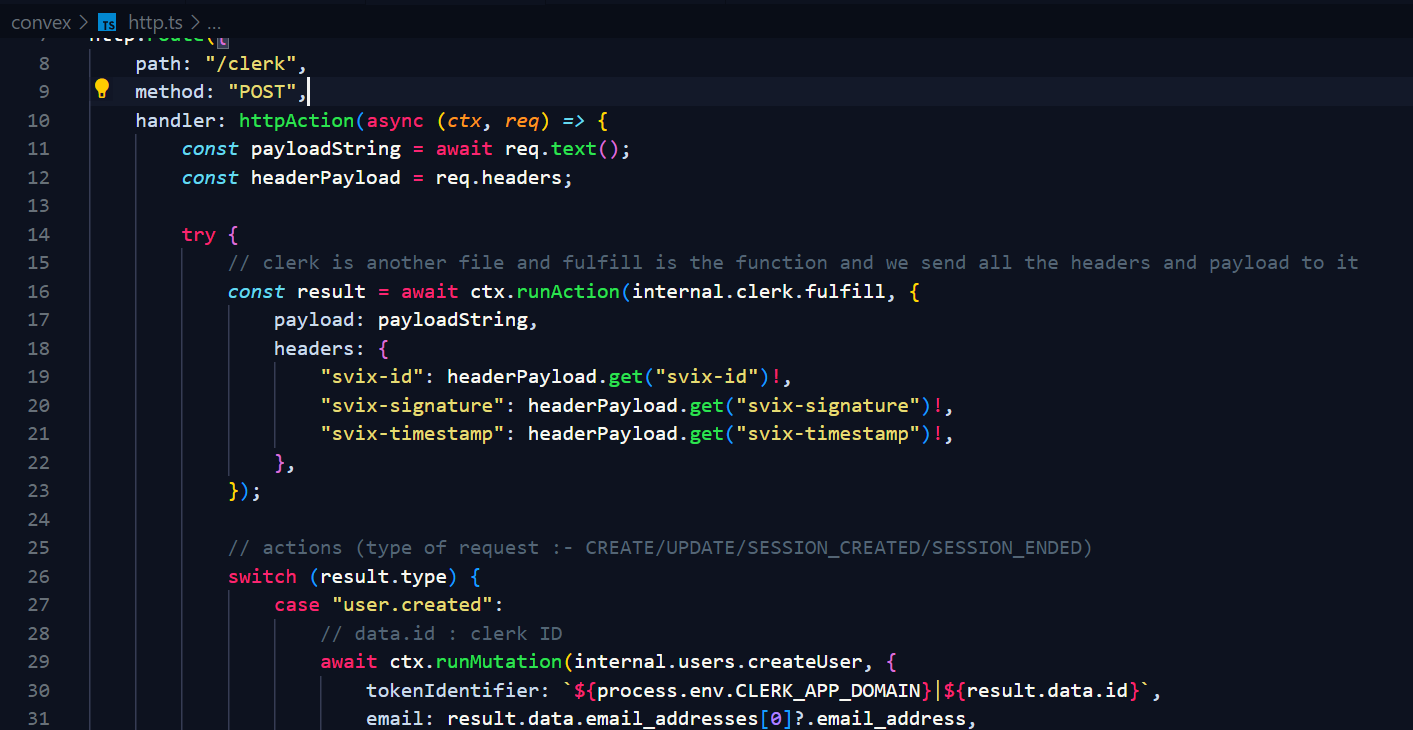
https://majestic-vole-679.convex.site/clerk



After creating the Webhook in the Clerk, copy the signing secret and create a ENV in Convex dashboard

A screenshot of a computer

Description automatically generated



**A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedExample of result variable data**

A screen shot of a program

Description automatically generatedA screen shot of a computer program

Description automatically generatedHere, clerk id is used for user creation and updating and

User\_id is used for session related task (user\_id is used only when type is session.created or ended)

For emojis:

npm install emoji-picker-react