



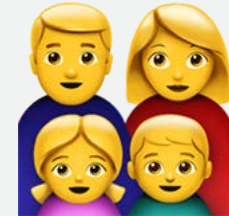
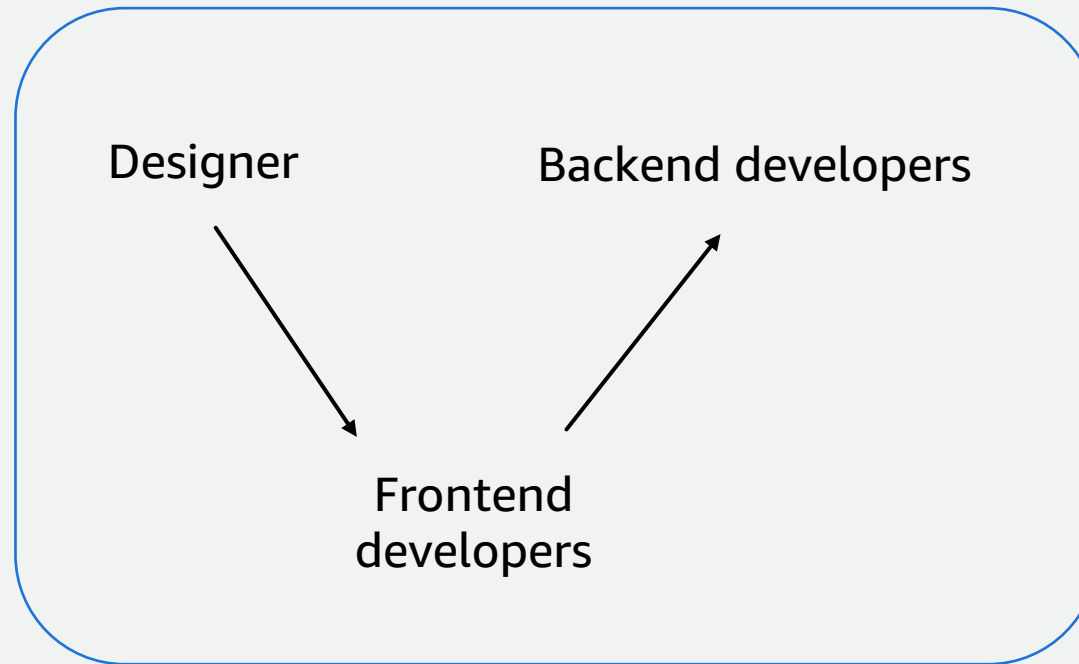
CONF42: JAVASCRIPT 2023

Turn on easy mode for full-stack development with Figma and AWS Amplify

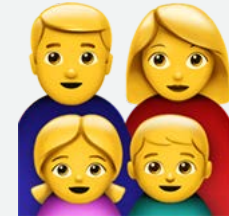
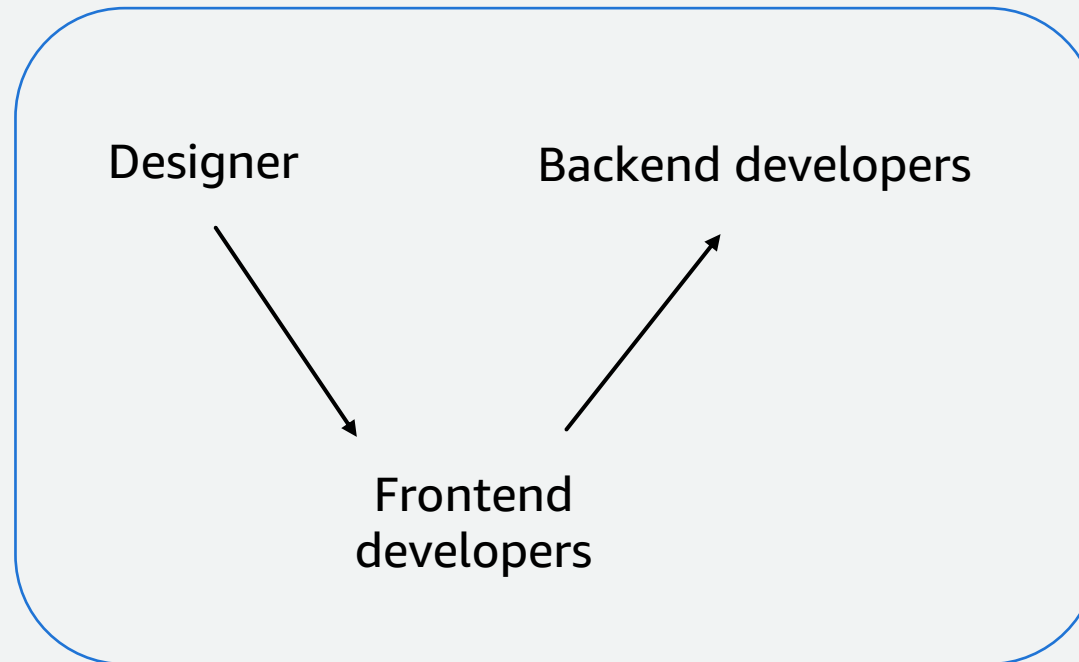
Arundeeep Nagaraj (he/him)

Manager, Solutions Architect - AWS

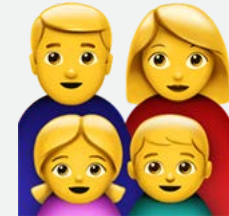
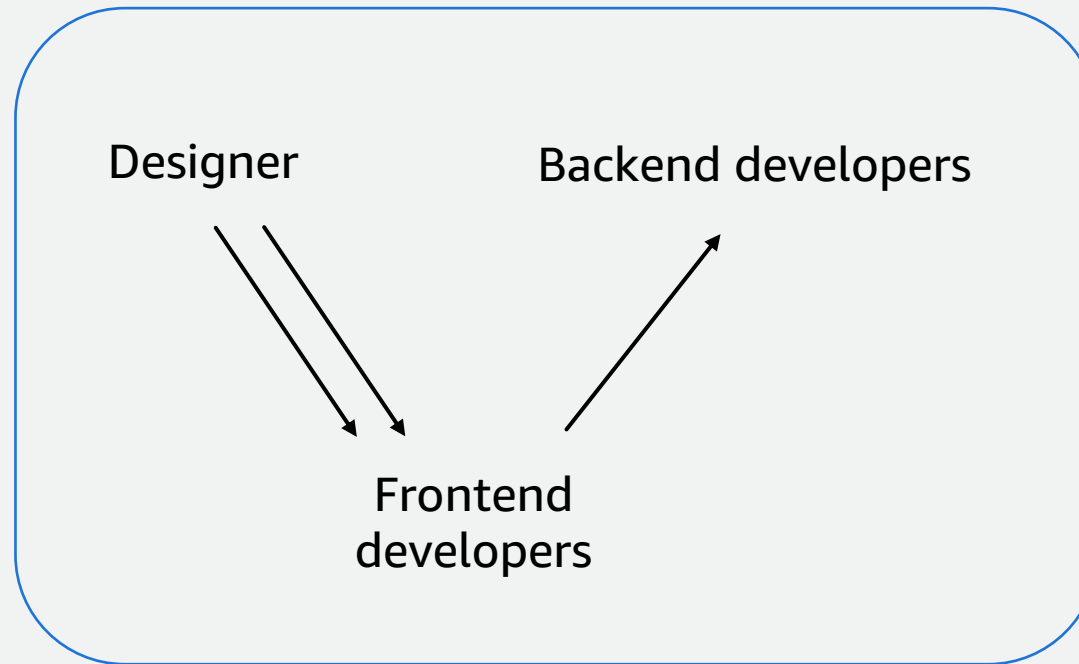
Design to application: Expectation



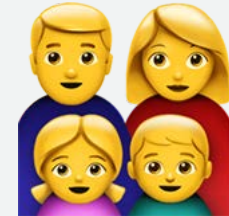
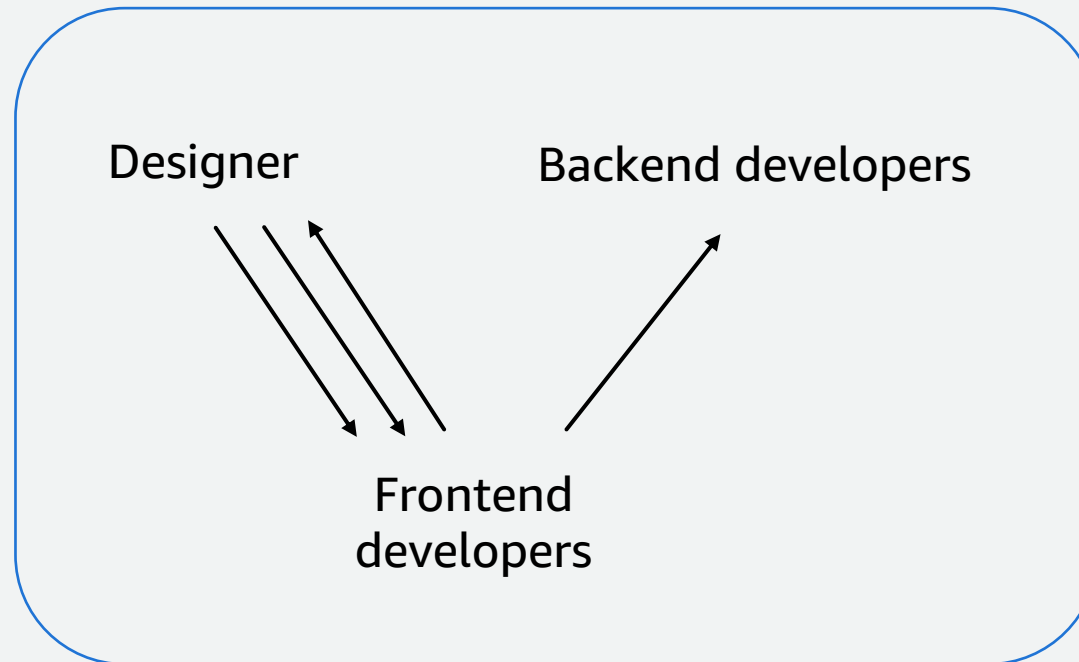
Design to application: Reality



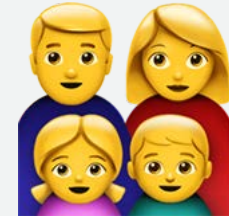
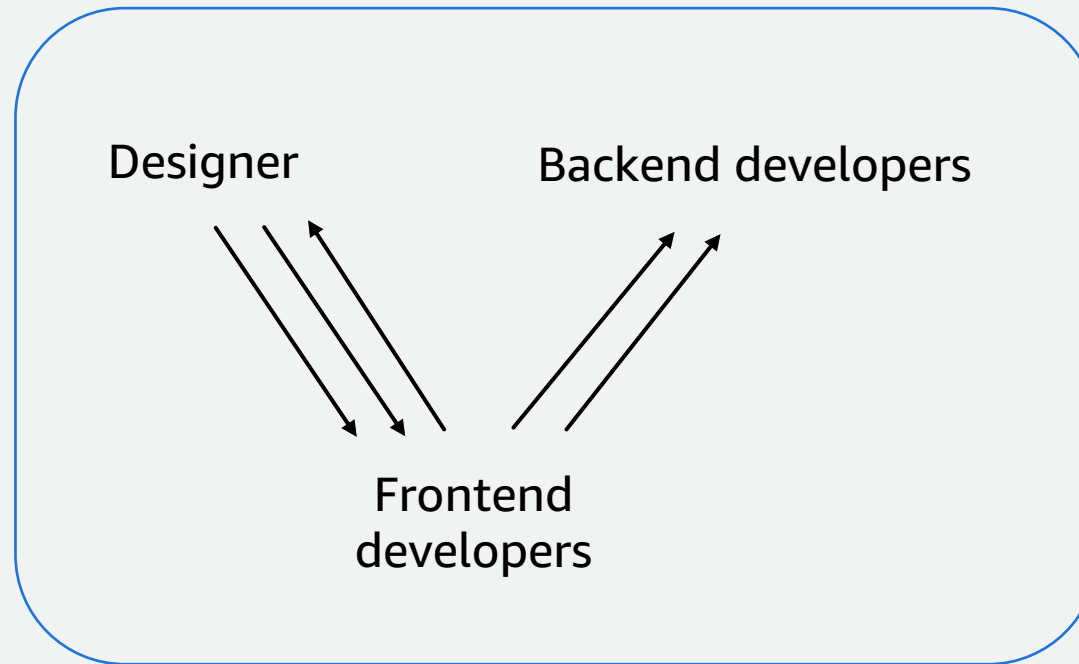
Design to application: Reality



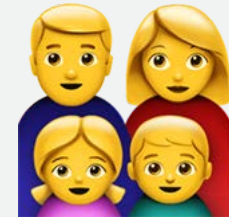
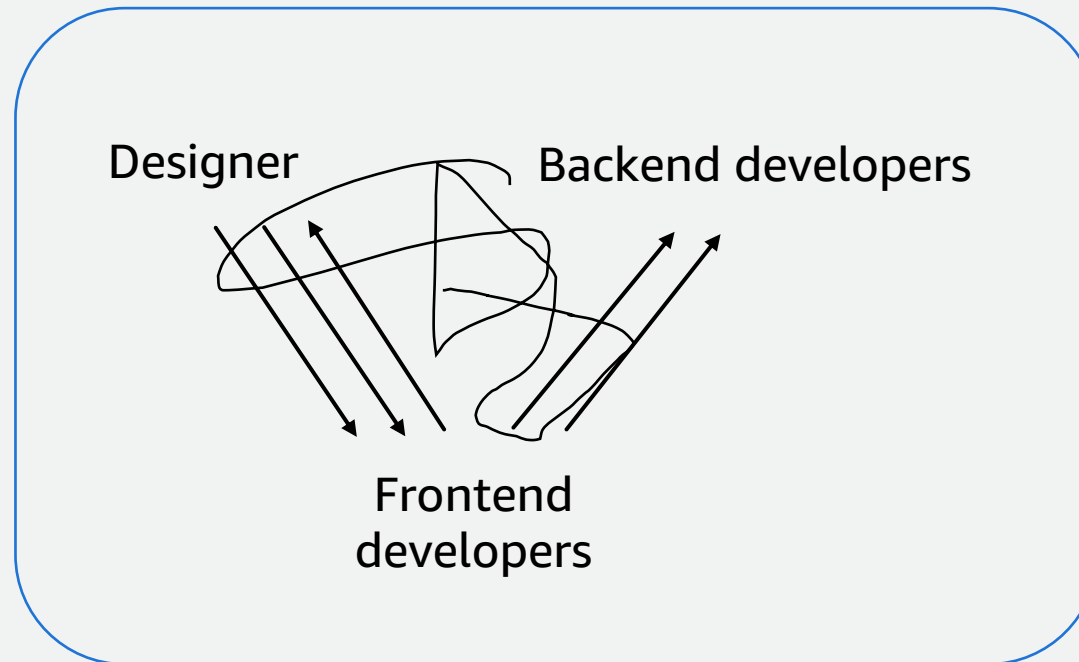
Design to application: Reality



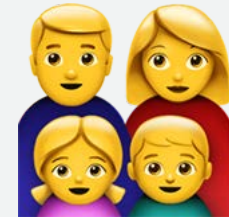
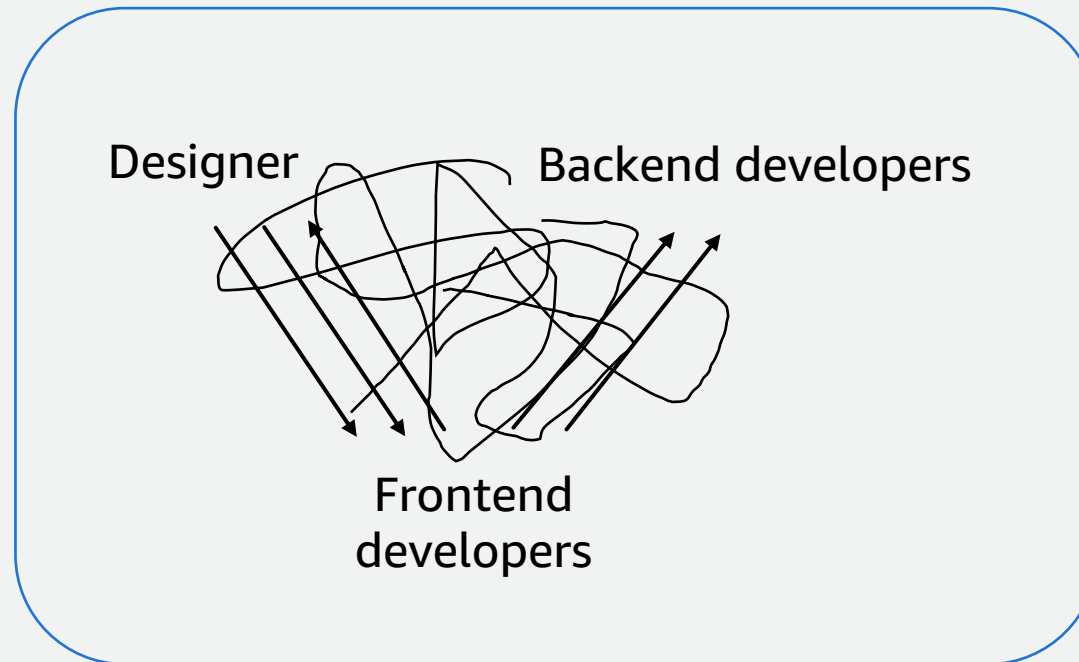
Design to application: Reality



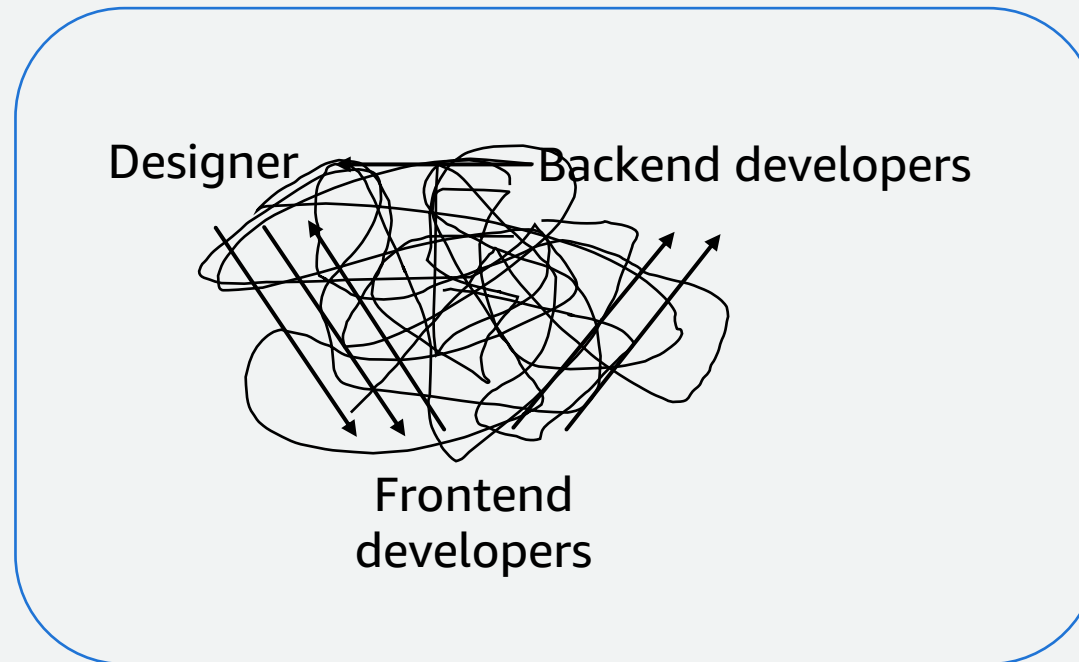
Design to application: Reality



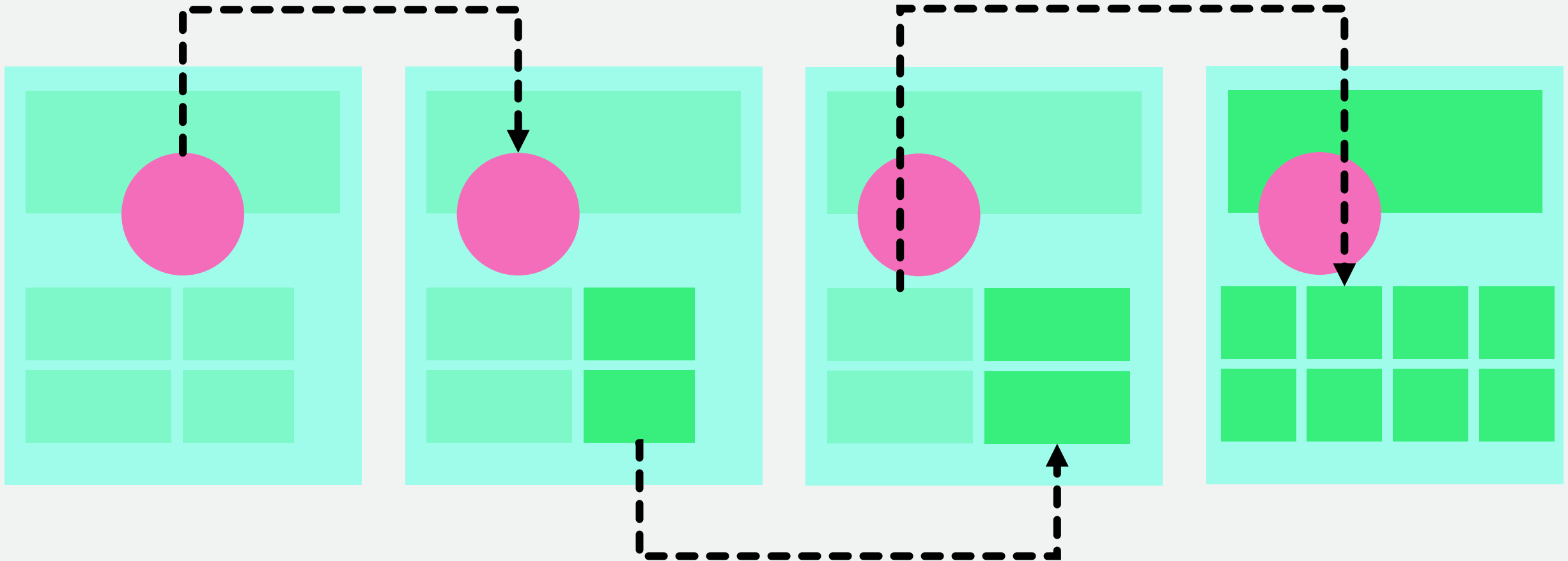
Design to application: Reality



Design to application: Reality



Developers spend months pixel-pushing



50%

of designers are not satisfied with
developers' implementation of designs

Designers not happy

Developers frustrated

What if . . .



AWS Amplify Studio

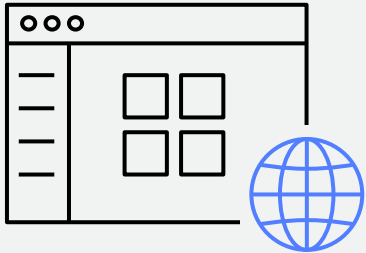


Amplify Studio

Build your app frontend with AWS Amplify

VISUALLY BUILD A FRONTEND UI

Saves time
Minimizes coding
Fully customizable

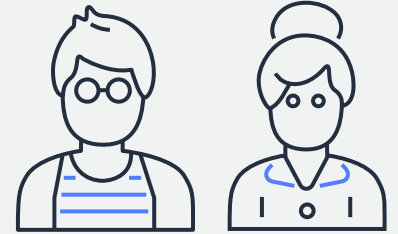


Choose from dozens
of open-source UI components



Figma

Build forms or convert Figma UX
designs to clean React code



Improves designer and
developer collaboration

Pixel-perfect designs with less
coding; maintain full control
over code

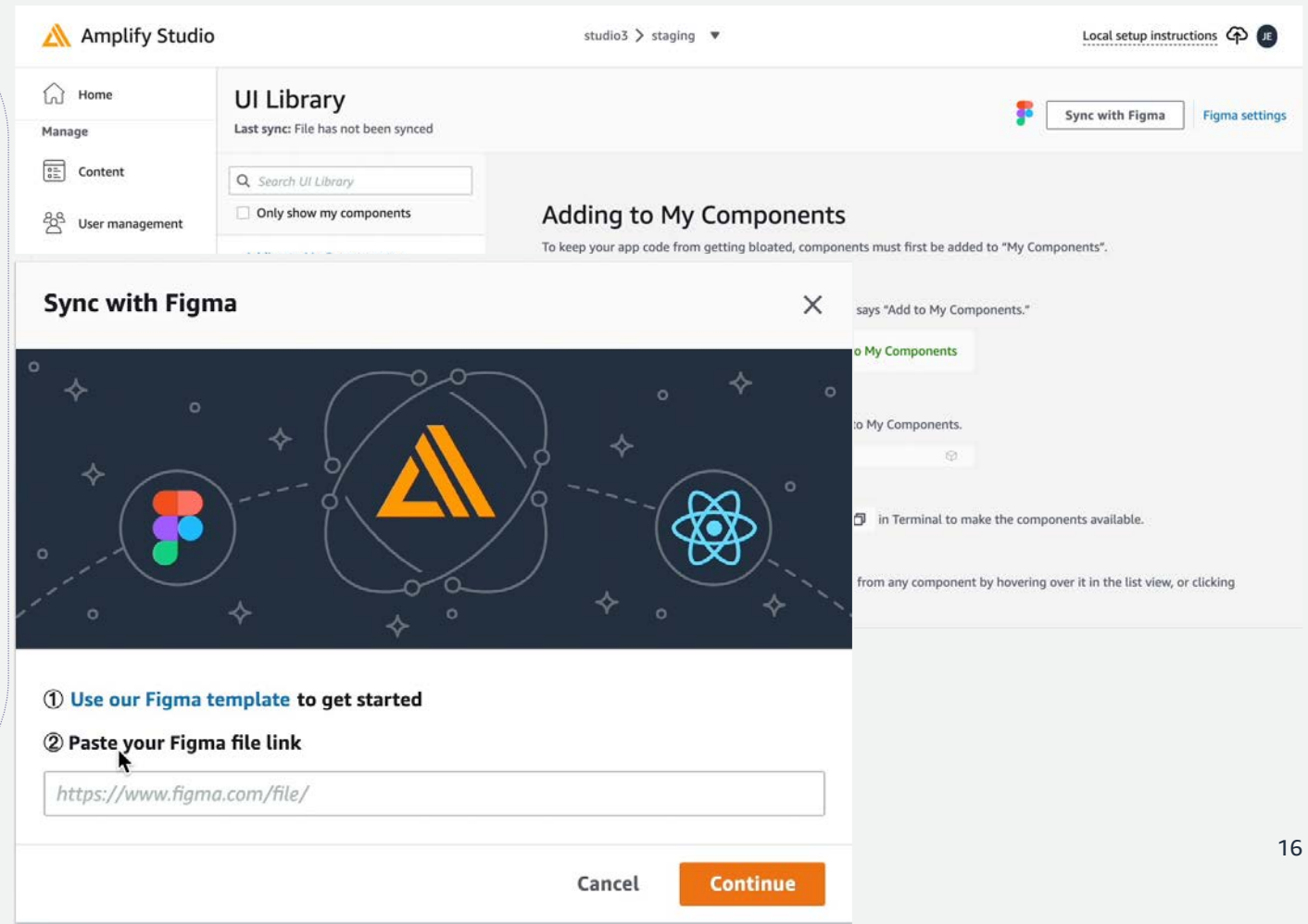
Build your app frontend with AWS Amplify

BUILD A FRONTEND UI WITH FIGMA-TO-CODE PLUGIN

Import UI component designs from Figma

Converts to clean React code

Work seamlessly with UX designers



AWS Amplify Overview



AWS Amplify

TOOLS AND SERVICES FOR FRONTEND DEVELOPERS TO BUILD WEB AND MOBILE APPS

Amplify Libraries



Use case-centric libraries to build cloud-powered mobile and web apps

Powered by AWS services and can be used with new backends created with the Amplify CLI and Amplify Studio or existing AWS backend

Amplify Studio



Visual interface for setting up your app backend

New: Visually build UI components and generate React code

Integrates with Figma to improve designer and developer collaboration

Amplify CLI



Toolchain to configure and maintain your app backend from your local desktop

Configure cloud functionality using the CLI's interactive workflow and intuitive use cases such as auth, storage, API

Amplify UI Library



Open-source design system with cloud-connected components for building feature-rich apps fast

Choose from dozens of prebuilt components that are themeable, accessible react components that connect to the cloud in a few steps

Amplify Hosting



Fully managed CI/CD and hosting service for fast, secure, and reliable static and server-side rendered apps that scale with your business

Supports modern web frameworks such as React, Angular, Vue, Next.js, Gatsby, Hugo, Jekyll, and more

Features you can build with AWS Amplify

DELIVER CLOUD-ENABLED EXPERIENCES BY INTEGRATING A BROAD RANGE OF CAPABILITIES INTO APPS



Interact with multiple data sources among AWS databases and microservices



Low-latency content delivery with minimum network requests & reduced payload with GraphQL



Flexible business logic with AWS Lambda and AWS Step Functions to interact with data and support your specific app workflows



Fully-managed user directory that scales to millions of users and supports sign-in with identity providers



Multi-factor auth, sign-up, sign-in, and forgot password to avoid building any boilerplate login workflows



Geospatial data, maps, and location search functionality in just a few lines of code



Offline use cases by storing data locally and automatically synchronizing with the backend when connectivity returns



Real-time interactive experiences using WebSockets to push updates to web and mobile clients



Stored user content such as photos and videos for your app in public, protected, or private storage buckets



Integrated push notifications and in-app messages for segment-based targeting, campaigns, and transactional messages



AI-powered chatbots in your web or mobile apps



Text translation between languages, text to speech, text from image, entities recognition, and more

AWS Amplify benefits

THE FASTEST WAY TO BUILD EXTENSIBLE, FULL-STACK WEB AND MOBILE APPS

Get to market faster

Libraries

Data modeling

Built-in backend code

Client-side data access code generation

Figma to React UI component code generation (new)



Build feature-rich apps

Sign-up and sign-in workflows

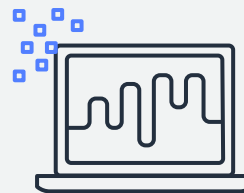
Location aware

Real time and offline

REST and GraphQL API calls

SQL, NoSQL, and cloud storage

AI/ML, voice, text, predictions



Scale

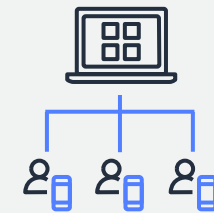
Global

Reliable

Secure

Scalable

Adaptable



Demo – Build a Cat Reviews application



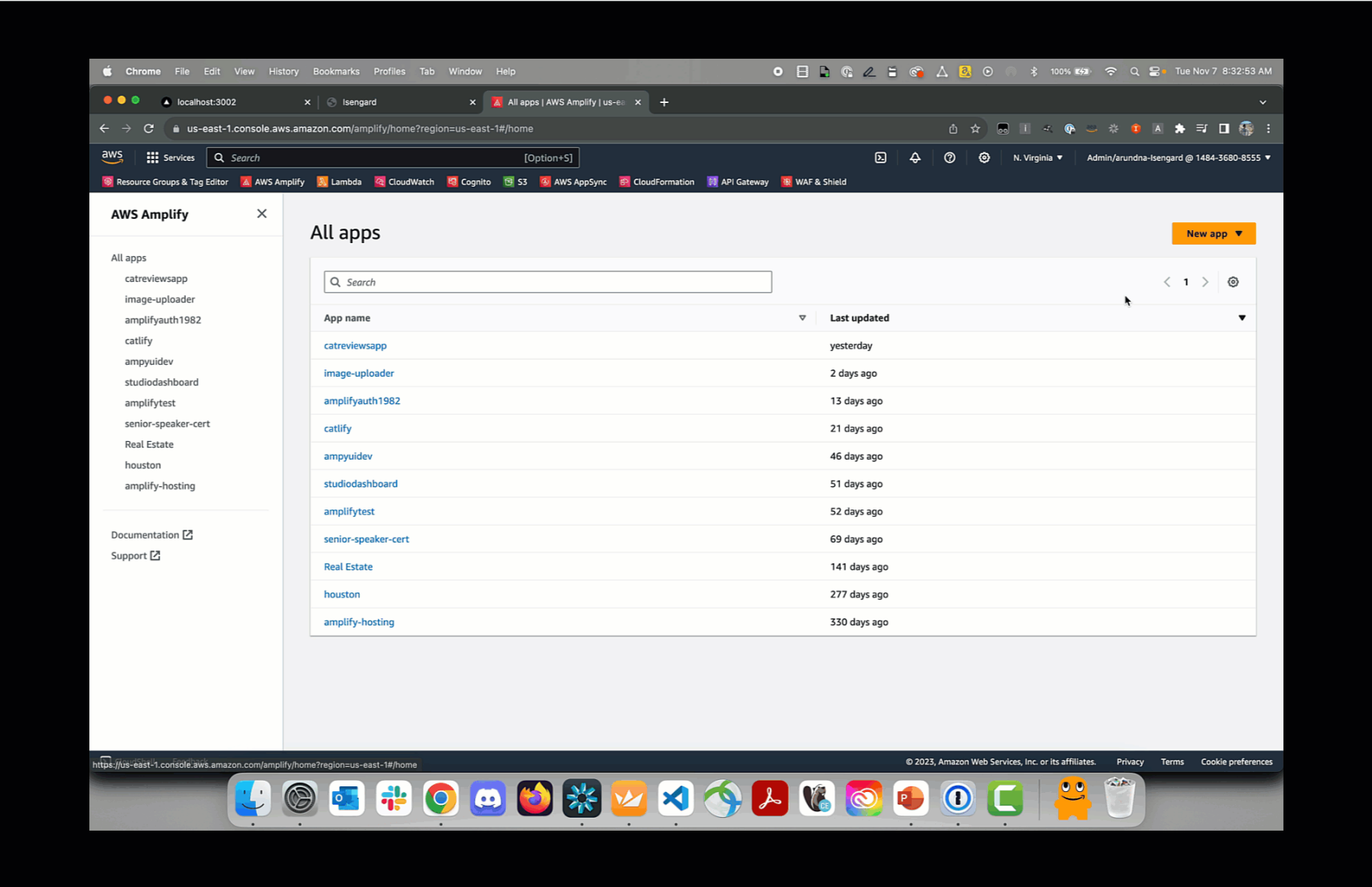
What we'll build

The screenshot displays a web browser window at `localhost:3002`. The interface is split into two main sections. On the left, there is a form builder titled "This is Amplify UI Form Builder". It contains the following elements:

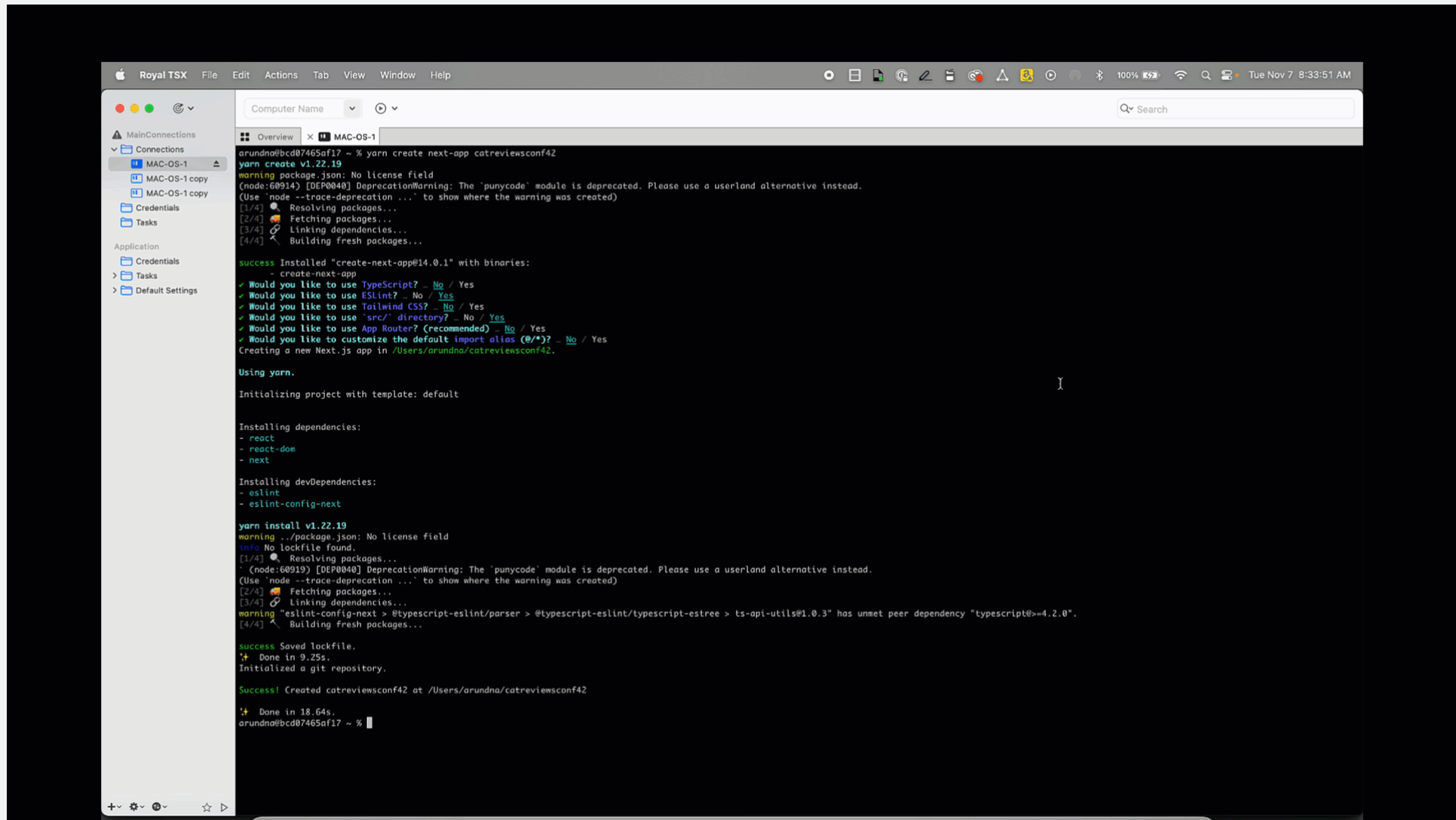
- Name:** A single-line text input field.
- Description:** A multi-line text area.
- Image:** A single-line text input field, likely for an image URL.
- Rating:** A range input with a minus sign, the number "0", and a plus sign.
- Buttons:** A "Clear" button and a "Submit" button.

On the right side of the browser window, the text "No results found" is displayed, indicating that no data has been retrieved or saved yet.

1 Build a new app with Amplify Studio



2 Install Amplify packages and build your data model



The screenshot shows a macOS terminal window with a dark background. The window title is "Royal TSX" and the menu bar includes "File", "Edit", "Actions", "Tab", "View", "Window", and "Help". The status bar at the top right shows "Tue Nov 7 8:33:51 AM". The left sidebar of the Royal TSX application shows a tree view with "MainConnections", "Connections", "Application", "Credentials", "Tasks", and "Default Settings". The main pane displays the output of a terminal session. The user is in a directory named "catreviewsconf42" and runs the command "yarn create next-app catreviewsconf42". The output shows the installation of Next.js 14.0.1 and the selection of TypeScript, ESLint, Tailwind CSS, and the App Router. The user then runs "yarn install" and "yarn install v1.22.19", which shows the installation of dependencies and devDependencies. The output concludes with "Success! Created catreviewsconf42 at /Users/arundna/catreviewsconf42".

```
arundna@bcd07465af17 ~ % yarn create next-app catreviewsconf42
yarn create v1.22.19
warning package.json: No license field
(node:60914) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Installed "create-next-app@14.0.1" with binaries:
  - create-next-app
✓ Would you like to use TypeScript? .. No / Yes
✓ Would you like to use ESLint? .. No / Yes
✓ Would you like to use Tailwind CSS? .. No / Yes
✓ Would you like to use 'src/' directory? .. No / Yes
✓ Would you like to use App Router? (recommended) .. No / Yes
✓ Would you like to customize the default import alias (0/*)? .. No / Yes
Creating a new Next.js app in /Users/arundna/catreviewsconf42.

Using yarn.

Initializing project with template: default

Installing dependencies:
- react
- react-dom
- next

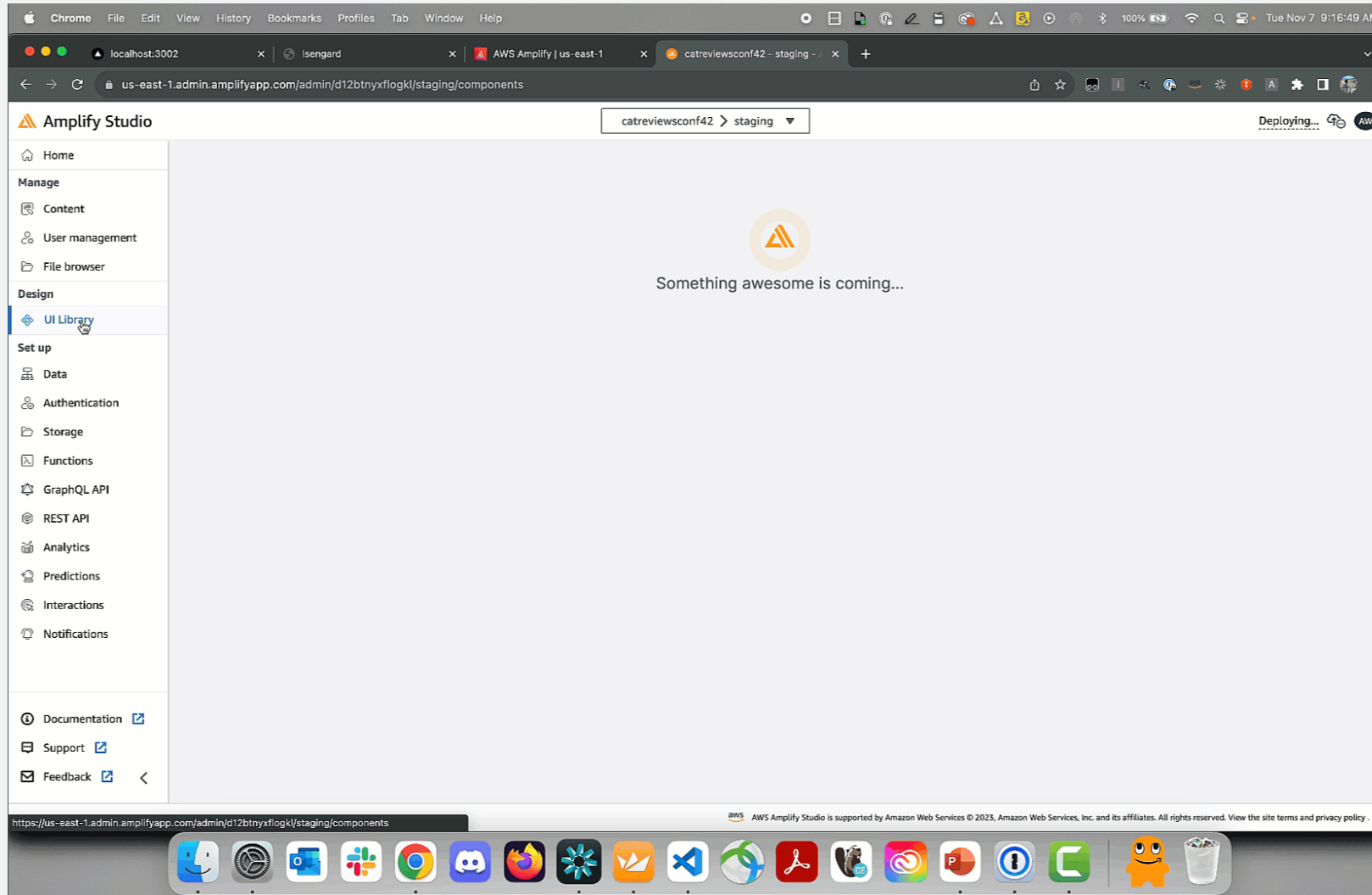
Installing devDependencies:
- eslint
- eslint-config-next

yarn install v1.22.19
warning ../package.json: No license field
info No lockfile found.
[1/4] Resolving packages...
(node:60919) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning "eslint-config-next > @typescript-eslint/parser > @typescript-eslint/typescript-estree > ts-api-utils@1.0.3" has unmet peer dependency "typescript@>=4.2.0".
[4/4] Building fresh packages...

success Saved lockfile.
* Done in 9.25s.
Initialized a git repository.

Success! Created catreviewsconf42 at /Users/arundna/catreviewsconf42
* Done in 18.64s.
arundna@bcd07465af17 ~ %
```

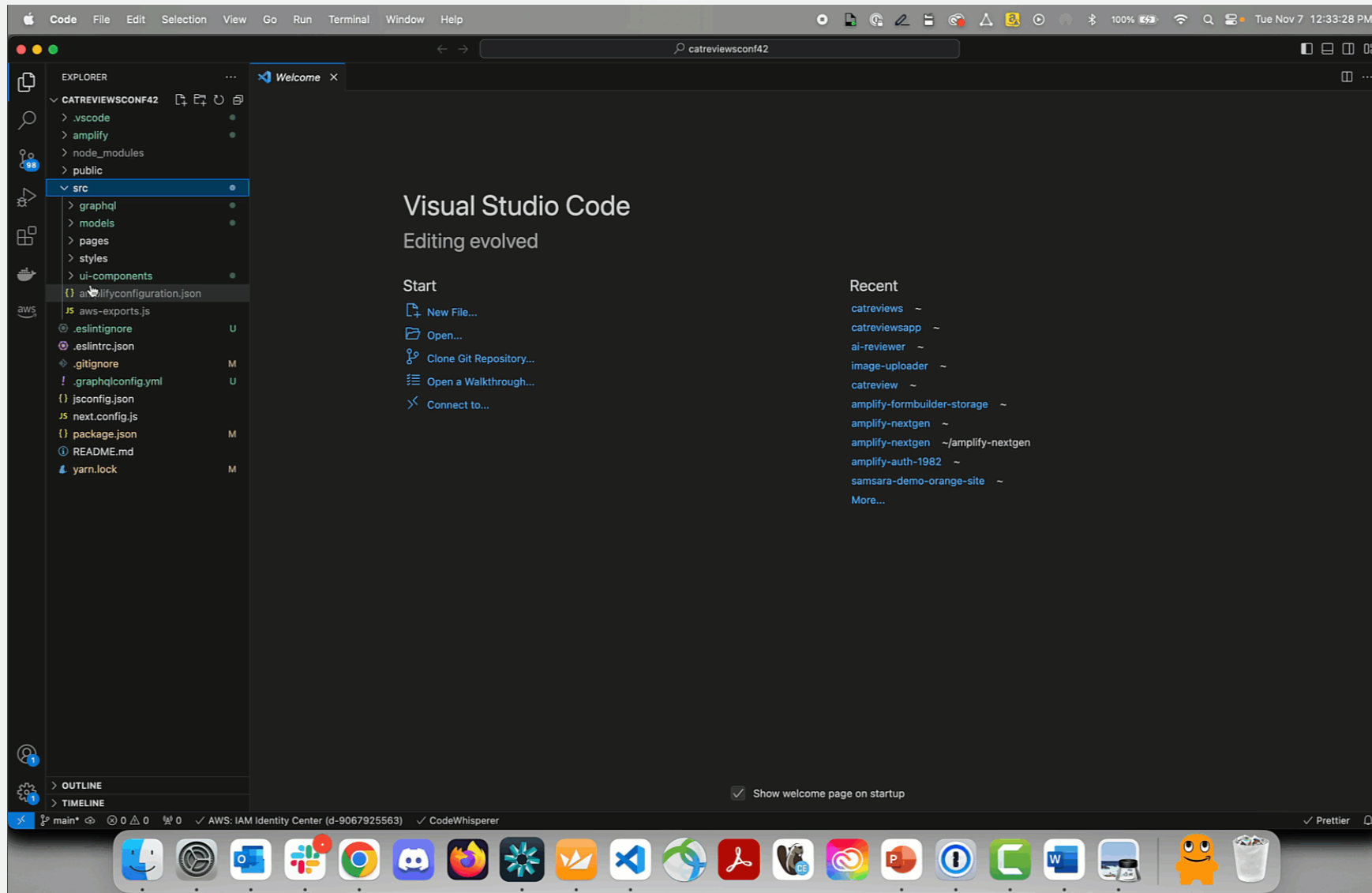

3 Design using Figma and bind data model



4 Configure Form Builder and setup local environment

The screenshot displays the AWS Amplify Studio web application. The browser address bar shows the URL `us-east-1.admin.amplifyapp.com/admin/d12btynxflogkl/staging/components/`. The interface includes a left-hand navigation menu with sections like 'Manage' (Content, User management, File browser), 'Design' (UI Library), and 'Set up' (Data, Authentication, Storage, Functions, GraphQL API, REST API, Analytics, Predictions, Interactions, Notifications). The 'UI Library' section is active, showing a list of components under 'Forms (2)' and 'Components (32)'. The 'ReviewCard' component is selected and displayed in the main area. A green banner at the top indicates 'Successfully synced with Figma' and '32 of 32 components updated'. The 'ReviewCard' preview shows a card with a heart icon, the name 'Monique Gottlieb', a paragraph of text, a 'New!' badge, five stars, and a 'Reviews' link. The interface also includes buttons for 'Copy code', 'Configure', and 'Delete' for the selected component.

5 Setup app code and create collection to render components



6

```

return (
  <ThemeProvider>
    <Flex direction="row" alignItems="flex-start" minHeight="100vh">
      <View flex="1" position="sticky" top="0" padding="large">
        <ReviewsCreateForm
          onSuccess={() => {
            showAlert("Review saved!")
            ref?.current?.clearFiles()
            setTimeout(() => {
              showAlert(undefined)
            }, 2000)
          }}
        />
        {alert ? <Alert variation="success">{alert}</Alert> : null}
        <Text color={tokens.colors.font.tertiary}>This is Amplify UI F
      </View>
      <View
        flex="1"
        alignSelf="stretch"
        backgroundColor="background.tertiary"
        padding="xl"
      >
        <ReviewCardCollection items={items}/>
      </View>
    </Flex>
  </ThemeProvider>
);
}

```

```
import ReviewsCreateForm from "@ui-components/ReviewsCreateForm";
import * as React from "react";
import ReviewCardCollection from "@ui-components/ReviewCardCollection";
import { API } from "aws-amplify";
import * as subscriptions from '../graphql/subscriptions';
import { Alert, View, Text, Flex, useTheme } from "@aws-amplify/ui-react";
import { ThemeProvider } from "@aws-amplify/ui-react";
import { listReviews } from "../graphql/queries";

export default function Home() {
  const [items, setItems] = React.useState([]);
  const [alert, setAlert] = React.useState();
  const { tokens } = useTheme();

  React.useEffect(() => {
    API.graphql({query: listReviews}).then(value => setItems(value.data.listReviews.items)).catch(console.log)
    const sub = API.graphql(
      { query: subscriptions.onCreateReviews }
    ).subscribe({
      next: ({ provider, value }) => {
        console.log(value.data.onCreateReviews)
        setItems(prev => [...prev, value.data.onCreateReviews])
      },
      error: (error) => console.warn(error)
    });

    return () => {
      sub.unsubscribe()
    }
  }, [])
}
```



Thank you!

Arundeeep Nagaraj

LinkedIn - @arundeeepn

X (Twitter) - @arundeeepnagaraj