

a dreamforce invite.

01 RED.UI

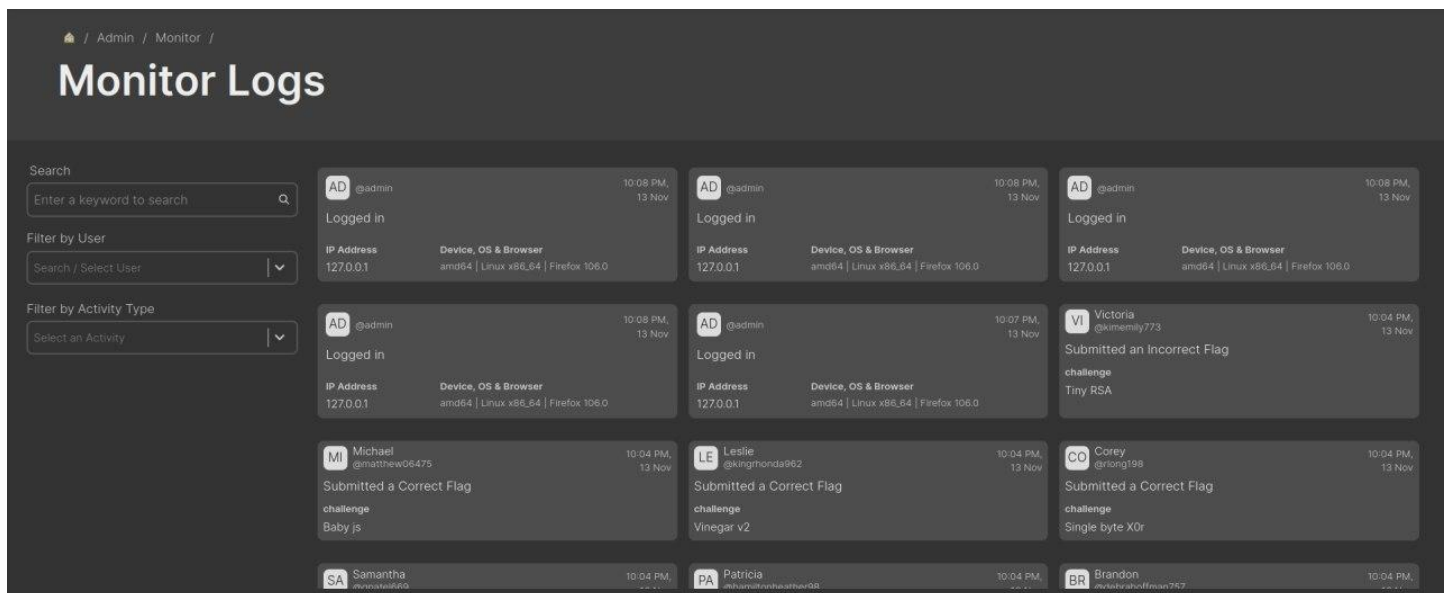
150 points

UI/UX

Here is a poorly designed page from an application. Can you make this log monitoring page look better?

This page is used to display logs from users both admins and users for the activities they perform on our CTF platform. The page is to be used by the organisers of the CTF, to analyse the CTF... who logged in when? did A solve before X? Did A copy from X? did both A and X submit the same wrong solution (flag) ?

There are several types of logs, each with different types of data. The user also needs to be able to search, filter, and sort the logs.



+ 50 points - Use a professional tool like Figma, Adobe XD

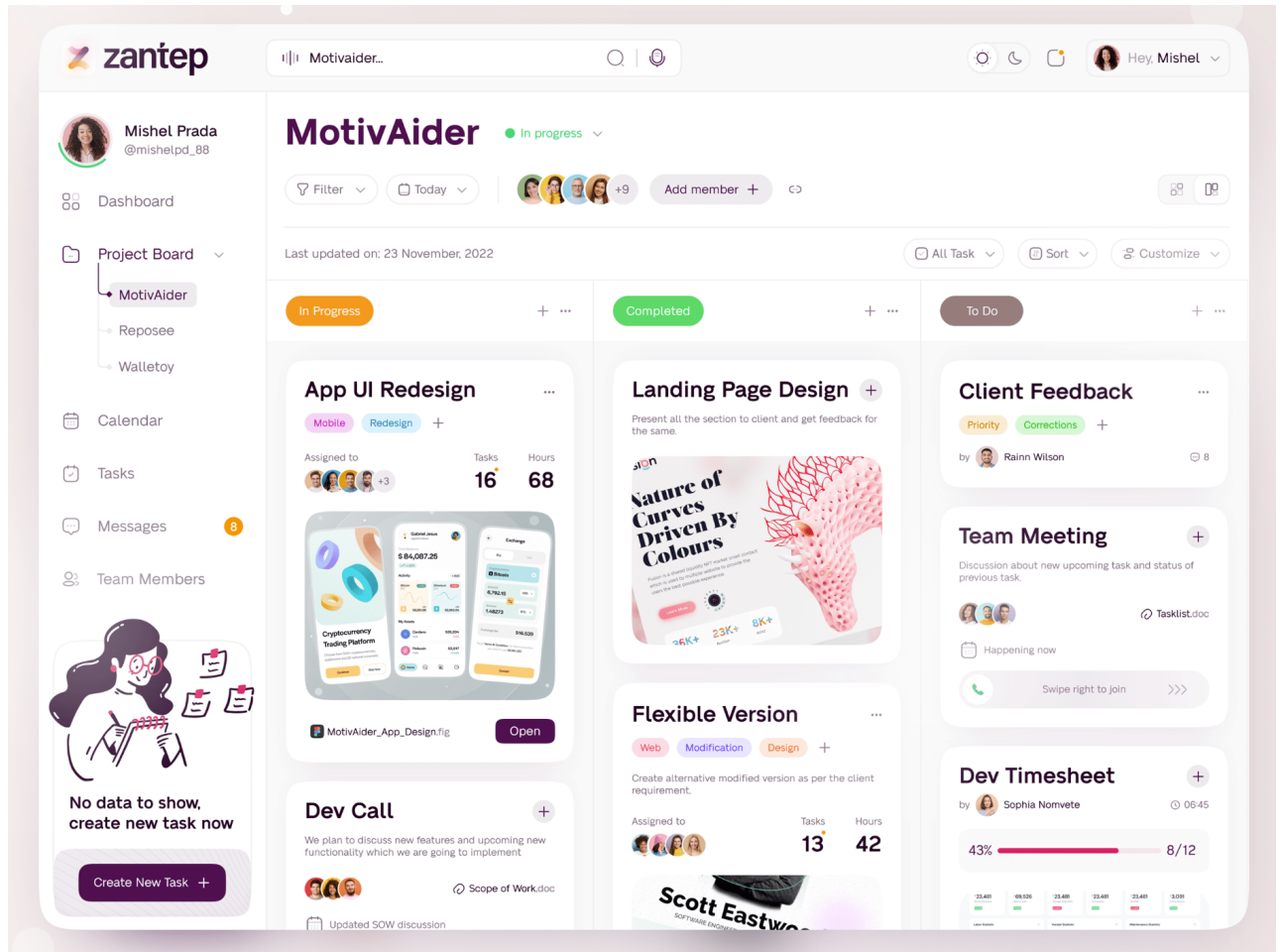
a dreamforce invite.

02 COPY.UI

Frontend

150 points

Build a webpage for the mockup given below, as perfectly matching as you can.



credits: [harshal patel on behance](#)

+ 150 points - use React/NextJS to build instead of HTML, CSS

+ 100 points - for mobile responsiveness

+ 100 points - Support for basic interactivity (with JS / React)

a dreamforce invite.

03 BDAY.CAL

Algorithm

100 Points

Do you know when you were born? Not by Gregorian calendar though.

Make a programme in any language that converts between a date in the Gregorian calendar to the corresponding date in Hindu calendar.

You should be able to print out the date, month, and day as per the Hindu calendar for a user inputted date as per Gregorian calendar.

+ 50 points - make a website for this

+ 50 points - deploy this website and share a link

04 HP.WIKI

API

200 Points

Make a Harry Potter character wiki using a NextJS (React) web application.

Use publicly available REST APIs to fetch data -

- <https://hp-api.herokuapp.com/>
- <https://github.com/theDavidBarton/the-harry-potter-database>
- <https://the-harry-potter-database.herokuapp.com/>

The application must have the following pages -

- / - A landing page with a google-like search?
- /characters - listing all prominent characters, with a search, and house-wise filtering, sorting by first name / last name ...
- /c/<name> - A profile for each character
- /house/<name> - A profile for each Hogwarts house, the profile should list

+ 100 points - for making the web app mobile responsive

+ 50 points - for deploying the app in Vercel

+ 150 points - for building a search suggestion (*autocompletion*) box

send as an invite. do not share (answers)

05 POKE.QL

Backend

500 Points

Design & develop functional GraphQL APIs for a (Pokemon) Pokedex application, using Python - Django - Strawberry. The application should be loaded with some pokemon data for demoing.

There should be atleast following APIs -

- pokemon(name: str, id: ID) → PokemonType
- pokemons(search: str, type: str) → Type[PokemonType]
- types(search: str)
- addPokemon(pokemon: PokemonInputType)
- updatePokemon(id: ID, update: PokemonInputType)
- deletePokemon(id: ID)

You (may) need to make Django models (Tables) to store the various details about the Pokemon. Learn about Django models & field types here -

- <https://docs.djangoproject.com/en/4.1/topics/db/models/>
- <https://docs.djangoproject.com/en/4.1/ref/models/fields/>

You can obtain data for the pokemons (and field names required for PokemonType) from PokeAPI - <https://pokeapi.co/>

You will be using Strawberry GraphQL Library to implement this. Read more about it here - <https://strawberry.rocks/docs/general/queries>

You do not have to make a front-end for this. Instead, we will look at your Django Admin panel and test your APIs through the GraphiQL playground.

+ 150 points - for implementing API keys (HTTP Header tokens) that are required to authenticate and access your APIs.

+ 150 points - supporting Pagination for pokemons Query

+ 300 points - for making a frontend webapp for this

06 DESIGN.DB

Database

250 Points

Design database schema (tables, fields and relationships) for an application to manage a Rubik's cube event, and provide an ER diagram of the DB and write SQL for the below requirements.

There will be different categories of events -
2x2, 3x3, 4x4, 5x5, Pyraminx, Mirror, Skweb etc. etc.

There will be different age groups of participants -
Under 6, 6-8, 8-10, 10-12, 12+, Parents & Mentors

Each participant's registration data should also be saved. The profile could have -
name, email, phone, date of birth, city, state, country, events (categories) participating, gender, mode of participation (online/offline)

For each category of event, the participant registered will have a score which is basically the duration the participant takes to solve the cube. (mm:ss:mm)

The following features / views would be needed, for which you need write an SQL query based on your DB schema -

1. View participants for a given age group
2. View participants for a given category
3. Search a participant using name or phone number
4. View scoreboard for a given age group for a given event
5. View top participant across events for a given age group
6. View top participants in each age group for a given event

+ 250 points (2x points) - for implementing the DB in Django, and writing Django queries for these views.

a dreamforce invite.

07 DEP.LY

350 Points

DevOps

Installing multiple operating systems on a single computer can often be full of hassle. Create and deploy a linux virtual desktop on the cloud which can be accessed from any device using a modern web browser.

Use AWS, Azure or GCP.

<https://www.youtube.com/watch?v=gsvS2M5knOw>

<https://guacamole.apache.org>

Components of the Project -

1. Web Browser - HTML5 Modern web browser which shall be used by the user/client to access the remote desktop
2. RDP Client & Gateway Server - Apache Guacamole is an open-source library that acts as a clientless remote desktop gateway. It connects to the remote desktop instances using RDP/VNC/SSH and relays back the shell or GUI to the client on the web-browser.
3. Remote Desktop Instances - Leased servers that run an operating system of choice, with or without GUI, with the Guacamole client installed. These are booted-up / accessed by the Guacamole server which establishes a remote access to these from the user's browser.

