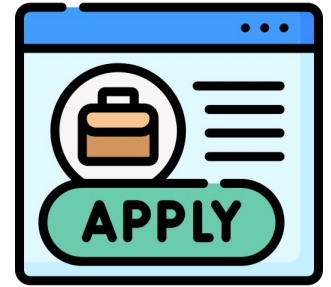


# JobSync

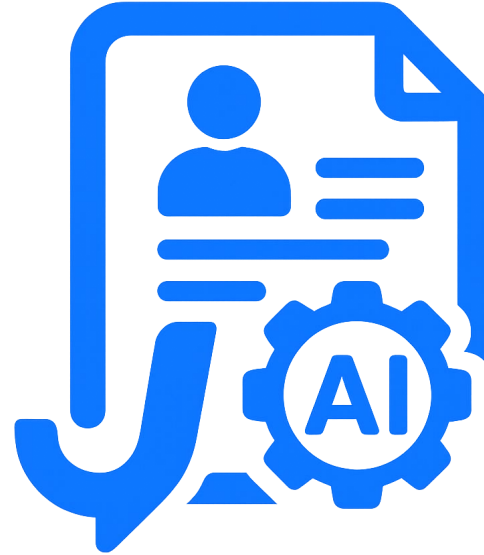
Jacky Wang, Miseok Kim, Jonathan He, and Sahil Patel

# Problem



## Solution: JobSync

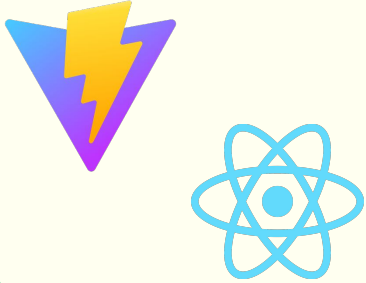
JobSync is an AI-powered job application assistant designed to streamline the process of **filling out job applications, crafting tailored resumes, and preparing for interviews.**



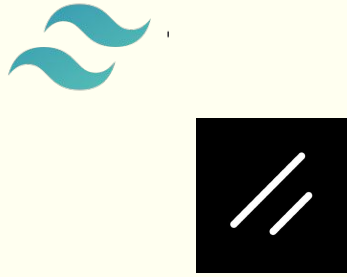
Live Demo

# Tech stack

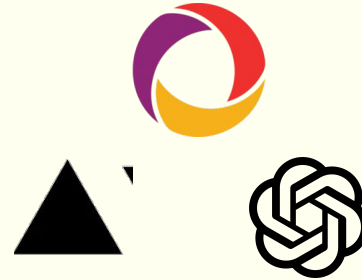
## FrontEnd



## Styling



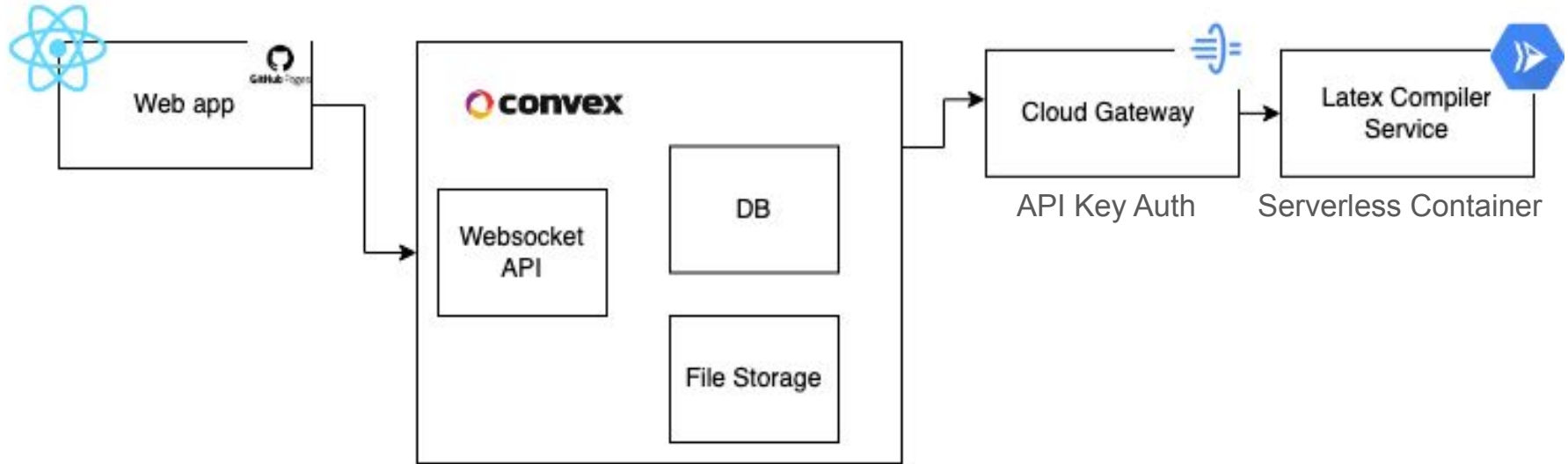
## Backend+Ai



## Deployment



# High-level Architecture



# High-level overview of AI features

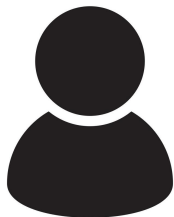
- Parse resume
- AI-generated Application Answers + improve answers
- Generate Interview questions
- Improve & grade user's answers to interview questions
- Generate sample answers to interview questions
- AI Chat
- Generate Tailored Resumes
- Generate Resume Insights
- Improve Resume Content
- Job Fit Analysis



# Resume Parsing



MuPDF



```
const { object } = await generateObject({
  model: openai("gpt-4o-mini"),
  prompt: systemPrompt,
  schema: z.object({
    name: z.string().describe("Full name of the candidate"),
    email: z
      .string()
      .nullable()
      .optional()
      .describe("Email address of the candidate"),
    phone: z
      .string()
      .nullable()
      .optional()
      .describe("Phone number of the candidate"),
    location: z
      .string()
      .nullable()
      .optional()
      .describe("Location or address of the candidate"),
    socialLinks: z
      .array(
        z.object({
          platform: z
            .string()
            .nullable()
            .optional()
            .describe(
              "Name of the social platform. Only include LinkedIn, GitHub",
            ),
          url: z
            .string()
            .nullable()
            .optional()
            .describe(
              "Full URL to the candidate's profile on this platform. Please",
            ),
        })
      )
      .default([])
      .describe("List of the candidate's social media profiles"),
    education: z
      .array(
        z.object({
          institution: z
            .string()
            .nullable()
```



## AI-generated application answers



## AI-generated job fit

- Based on job details and user profile
- Makes a call to OpenAI as expert career advisor
- Outputs a string

# AI-generated interview questions and answers

## Interview questions

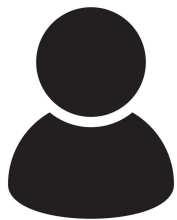
- Based on job details
- Makes a call to OpenAI as expert career advisor and interviewer
- Outputs JSON object with two arrays

## Interview answers

- Based on user profile
- Makes calls to OpenAI with specific prompts
- Outputs a string



AI Chat 



Gain confidence in your application

# Generating Tailored Resume



- Fetches full user profile and job from db
- Use structured output to generate a JSON structure of resume content
- Feed the content into a pre-existing latex resume template
- Invoke a separate latex compiler service to compile into PDF and display in UI

# Closing Remarks

## Limitations

- Heavily dependent on OpenAI APIs
- AI answers are not always reliable
- AI knowledge limited to user's profile (user's resumes), hallucinates responses otherwise

## Future

- Open source for the benefit of the community at JHU
- Integrate with other related projects
- Polish & advanced features (e.g. Chrome extension / autofill)

## Key Takeaways

- AI enables us to make powerful applications