ResumeUp

Final Report

Client: Elmin Didic

Advisor: Dr. Mitra

sdmay24-05

Team Members:

Nedim Hodzic
Elmin Didic
Nicholas Thomas
Mitchell Talyat
Nathan Boldt
Siddharthan Prakash

Email: sdmay24-05@iastate.edu

Website: https://sdmay24-05.sd.ece.iastate.edu/

Table of Contents

1 Introduction and Background

1.1 Problem Statement

In today's world, an effective resume is necessary to find any type of success in the job market. However, as many college students have found out, creating a concise yet detailed resume is almost impossible without putting in countless hours of work critiquing each section of their resume.

Once you account for the constantly changing standards of a 'good' resume, the simple task of creating a resume may seem like a daunting task. Our team is focused on making a website that can help create resumes while using artificial intelligence to make decisions based on the current standards of resume building.

1.2 Intended Users and Uses

Anyone looking for a job and in need of a strong resume will benefit from ResumeUp. Every job seeker will benefit from our AI-powered tool. The users will be able to input their information and an AI will help improve some of it, giving more detailed descriptions.

Use Cases:

- **Resume Optimization:** AI will take basic information for some sections, like the work experience section, and fill the resume in with more detailed information. This will make resume creation much faster and easier, giving the user powerful and descriptive resumes with only a few words.
- **Resume Viewing:** The user will have the ability to create their own personal account. With this account, any of the resumes they make will be saved to be downloaded again and again when they need it.
- **Formatting Assistance:** ResumeUp will format users' resumes by taking their information and filling out a template. This will give the user a clean and professional document without any of the hard work.

1.3 Context of Implementation

ResumeUp can be used online through a web browser. It should be used to create a new resume without the need to fill in some of the more tedious steps, like work descriptions. There are similar products, but ours caters to the software engineer/ ISU student locally.

2 Revised Design

During the actual development of ResumeUp, we noticed our requirements changing and evolving over time. Whether due to time constraints, waiting on OpenAI access, or our needs changing as development progressed, our functional and non-functional requirements are not the exact same as 491. Requirements like the resume grading requirement were changed a little due to the difficulty of implementing, or the requirement of listing feedback for each line of a resume was removed because of time constraints and ambiguity of what makes a resume line good or bad. Here we will go over our new requirements.

2.1 Functional Requirements

- Users can fill in their information, creating a new resume with AI assistance
- Users will be given a resume grade based on the information they provide
- A user can create an account and access that account
- A user can save a resume to their account and access it later

2.2 Non-Functional Requirements

- A user should be able to navigate the site and get through everything in under 2 minutes
- The website will be hosted on a server with at least 90% uptime
- The website will have 0 actions that cause the webpage to lock or freeze due to client-side operations
- The website should respond to user input within a few seconds

2.3 Engineering Standards

- **IEEE 828-2012 Configuration Management in Systems and Software Engineering**: This standard provides some requirements for configuration management. Since we are developing software, we need to ensure it is maintained properly and performs as expected over time, meaning we need good configuration management. Following these standards should help us achieve that.
- **29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering**: This standard will serve as a guide for good requirements definition on the software and how to apply these requirements effectively on the web application.
- **W3C Standards for Web Design and Application and Web Architecture**: This is a helpful resource for standards or recommendations for web development. The standards/recommendations regarding mobile web interface, APIs, HTML, CSS, and HTTP will be helpful for this project.

2.4 Security Concerns and Countermeasures

Since we are storing user data, there is always the concern about their information like passwords being leaked or their account being accessed without their permission. We also need to make sure our data is secure; things like our database access keys or our ChatGPT API key need to be hidden from the user.

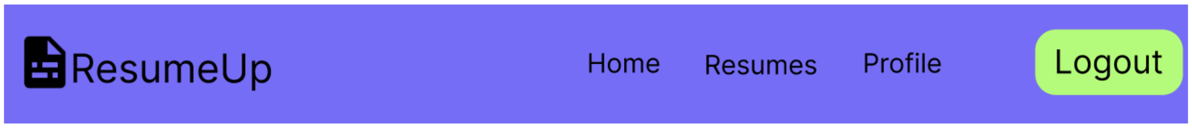
To remedy these, we have implemented the standard log-in and sign-up functionality seen on all kinds of websites. We make sure that when logging in, the password inputted matches the user's password to the exact character, and when signing up, no accounts can be re-made using the same email. To store our private keys, we ensure that none of our keys are hard-coded in the application code and are rather stored in environment variables.

2.5 Evolution of Design

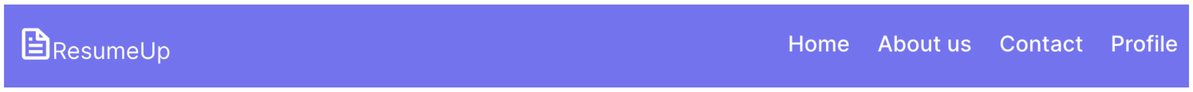
Our final design followed our original Figma pretty closely, with a few changes here and there. Things like the Terms of Service page, Home Page, and Contact Page followed the

Figma almost exactly, while things like the Header and Footer experienced some minor changes.

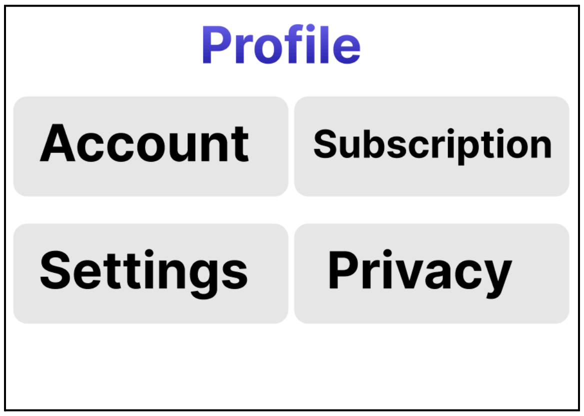
The major changes were things like the Resume Creations and Profile pages. During development, we decided that our original Figma design needed improvements for usability and visibility. Changes like displaying a user's resume on the profile page rather than a separate page were done to enhance the user experience. We feel our final design more accurately reflects our goals for ResumeUp and gave us a better final product in the end.



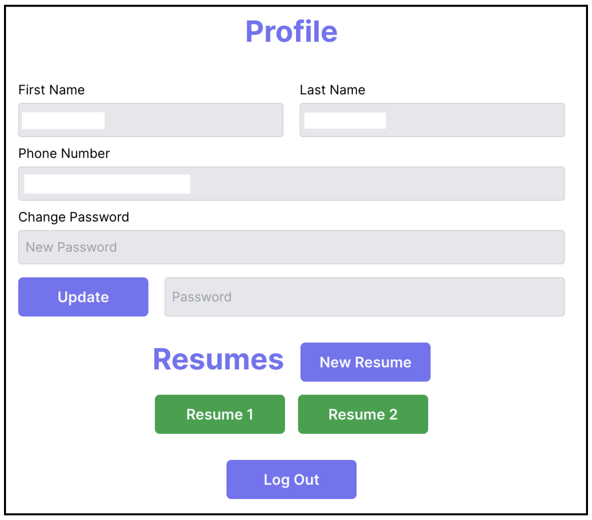
Original Header



New Header



Old Profile Page



New Profile Page

3 Implementation Details

3.1 Detailed Design

For the development of ResumeUp, we followed our original plan from 491 and used Next.Js with TypeScript. This allowed us to have an "all-in-one" development experience, with both our backend server-side and frontend client-side being in the same app. It also gave us an opportunity to learn using Next.Js and TypeScript which was something we all wanted out of ResumeUp - a learning experience. We wanted to be able to gain a strong understanding of new technologies through the development of ResumeUp. This came with some challenges like separating the server side and client side. Typically, these are already separated, and since they are together in Next.Js, there was a bit of a learning curve to using this paradigm.

We utilized React.js for the frontend of our website. This allowed us to make client components we could reuse throughout the website. Things like the header and footer were only made once, and any edits made only had to happen in one place. For our backend, we originally planned on using Express.js, but since Next.js has the server-side built-in, we did not have a need for it. Instead we used plain JavaScript when creating all the APIs needed for ResumeUp.

For our database needs, we decided to go with MongoDB Atlas. This differs from our original plan of using DynamoDb and we will discuss more on why we made this change in section 3.2. Like many cloud databases, using MongoDB gives us access to a scalable database that we can increase or decrease the capacity of based on need. This was great for initial development and testing because we did not have to worry about paying for a database to start off.

3.2 Functionality

We took time and care to ensure that ResumeUp functions like any other website a user may visit. We ensured that the website was responsive to clicks and that the user was redirected to the right place and at the right time. All functionality like resume creation or signing up works as you would expect it to, with responses given back to the user when something is successfully or unsuccessfully done. We also feel that the overall look of a website is a big part of the functionality, so we took care when creating all the pieces a user will see. We made sure that items were properly spaced out so the user could easily navigate each page, and we ensured our color scheme was simple and did not clash. By using colors that all work together, it creates a better viewing experience for the user, making them more likely to return to ResumeUp in the future.

We created our own account functionality rather than using premade libraries. This includes things like creating an account or displaying a user's information. We decided to code our own functionality as it would give us experience developing this sort of thing, allowing us to use it for future needs. Using an already existing library also did not make sense as some portions, like logging in, are simple to develop, and other portions, like displaying a user's resume, would most likely not be found in a library.

For the actual resume creation functionality, we utilized OpenAI's ChatGPT API for the AI functionality. This was utilized due to its ease of use and how good of a tool it is. All of us have used ChatGPT already for our personal resumes, so implementing it as our AI model made the most sense. OpenAI also provides ample documentation on setting up and using the API, so incorporating it was almost no trouble at all.

3.3 Implementation Notes

At the start of implementation there were a lot of things we wish we could have done a bit differently. One of those was the use of AWS Amplify for things like our database. This is a tool that we found that is great for web applications like ours but we spent too much time trying to understand how to use it. Eventually, we realized that it was not worth it to try and force the use of AWS Amplify as we would have almost no time left to actually develop ResumeUp.

We also spent a lot of time waiting on OpenAI access. Since ChatGTP tokens are not free we needed to wait for access to it from ETG. This took a bit of time and really hindered our development momentum as we had to wait a while to actually begin implementing ChatGPT into ResumeUp.

We decided to go with MongoDB Atlas for our database over DynamoDB for a few reasons. One of those is that it is initially completely free. There was no need to input a credit card, and we were able to start using it right away. The free tier was more than enough for our development and testing needs, and since MongoDB is highly scalable if there were ever a need to get more storage, it would not be difficult to do so. Another reason was the ease of use. We realized early on that trying to use brand-new technologies for every aspect of the application was going to be difficult. Some of us have used MongoDB before so we decided we would sacrifice some learning for quicker development. This gave us the opportunity to start implementing right away. Things like the login and signup backend were done early on, which saved us a lot of time when we started developing the more complex portions of the app. MongoDB also has lots of great documentation and user support, making it so any issues we came across were easily resolved.

4 Testing

4.1 Process

The tests that we executed on the project were manually tested on local hardware to ensure that it was functional and acting as intended. As an additional measure, we made sure that we had multiple developers review each pull request thoroughly before allowing it to be committed to main. One last testing measure we took was taking advantage of potential users to have them test our software. Friends and family tested ResumeUp, and we were able to take their feedback into account.

The frontend user interface, as well as the backend code were tested for their functionality. The functionality tested was many of the core features of ResumeUp. Some examples are as follows: resume creation, interactions with the AI API, user account creation, and log in. It was done to ensure that the user experience was smooth and worked as intended. Each feature was tested as they were implemented and changed. This was done before and after the pull request into main.

4.2 Results

Our results of testing gave us stronger insight into how a user might use the resume creation. We also saw how a user interacts with our user interface and could take that into consideration when working on fixing old features or creating new ones. Originally, our template form had poor UI, and we noticed test subjects would not fully complete it. From this, we made it easier to interact with the form.

We decided our results were sufficient after the feedback was considered and implemented, as well as functioning properly. After receiving any feedback, our team would discuss it to ensure that the results from the feedback were achievable and made sense.

5 Broader Context

5.1 Public Health, Safety, and Welfare

With our final design, the influence on the public safety and welfare of users is very minimal and was the same as our initial thoughts. Some potential areas of concern are data privacy, bias from the large language model, or accuracy of the results. User data could be hacked if stored incorrectly, and the results from the AI may not accurately reflect the user's intentions. With proper implementation, however, these issues can be avoided.

5.2 Global, Cultural, and Social

ResumeUp has an impact on a few groups, some of them being students or people in corporate jobs. People in these communities are typically in need of a resume and change their resumes often so having ResumeUp as a tool can help them immensely. We also feel that we represent the values of the members of these communities very well with ResumeUp. ResumeUp does a good job of reflecting what people in these communities would want out of a resume, and our use of ChatGPT closely matches the content someone would want in their resume.

5.3 Environmental

Our final design has a small, indirect impact on the environment. Since ResumeUp is a website it needs electricity to run the site and it needs something to access the site like a laptop or computer. If many people start to use ResumeUp, this would cause more devices to be in use, which all need electricity to run. If the demand for ResumeUp increases, there will also be an increase in cloud computing resources needed, which can have an effect on the environment. Say there is a need for another server to be used; That is another thing that is constantly running. We are also dependent on the cloud provider adhering to environmental practices, if they do not follow these then ResumeUp would indirectly be harming the environment.

5.4 Economic

We originally thought ResumeUp would have little to no economic impact but we now know that is not the case. Having to host the website, use a database, and use the ChatGPT API are all things that have the potential to cost us money. The ChatGPT API also has impacts on the user. If we were to make the creation of resumes completely free to every user, users could abuse this system by creating thousands of resumes, all of which use the ChatGPT API. This either puts us at risk of having extremely high costs or putting our services behind a paywall. Using ResumeUp also benefits the user by helping them get new, higher-paying jobs that they otherwise would not have gotten if they did not have a proper resume.

6 Conclusions

6.1 Progress Review

Our final result differed from what our intended goal was. We underestimated how long it would take to get university access to funding. Along with that, the creation of actual resume templates took a bit more time than we were expecting. We felt that this was probably for the better because we were able to make a more complete project for one certain area instead of having multiple parts that were not completed. We needed better time management, especially when it came to ETG. Our group had to focus on other classes as well, but we should've allocated a bit more time to actual development.

6.2 Implementation Value

The project creates powerful resumes for the users by utilizing ChatGPT to give the user enhanced information with more detail for their resume. This helps solve issues of having descriptive bullet points on a resume and will help them get a better job than they otherwise would not have got without a good resume.

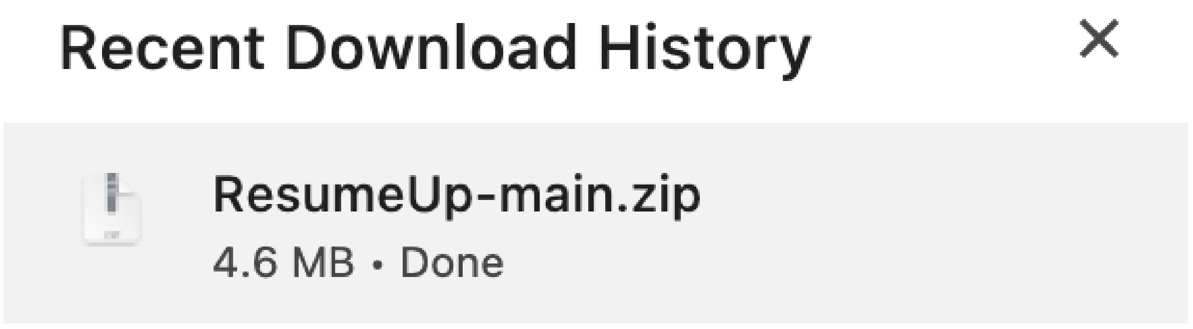
6.3 Future Steps

For the future of this project we plan to improve the AI's response, along with allowing users to edit their resumes. We also plan on adding more resume templates to give users more variety in the style of their resume. We also would like to add a bit more error checking for incorrect outputs from the AI.

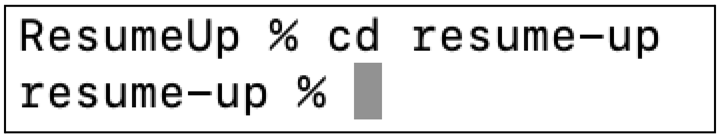
7 Appendices

7.1 Appendix 1 - Operation Manual

To start off, you will need to install the source code found in 7.4 Appendix 4 and unzip it.

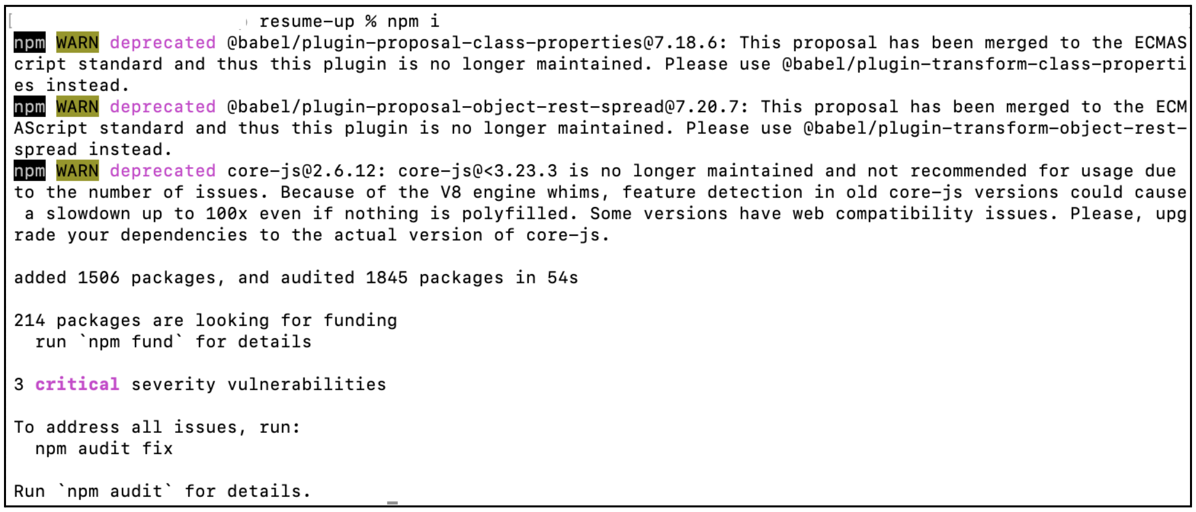


Next, you will need a terminal to run some commands. This could be a terminal provided by your computer or one found in some IDEs like VisualStudio Code. First, you will need to change directories in the "resume-up" directory. This can be done by entering the following command: "cd resume-up/":



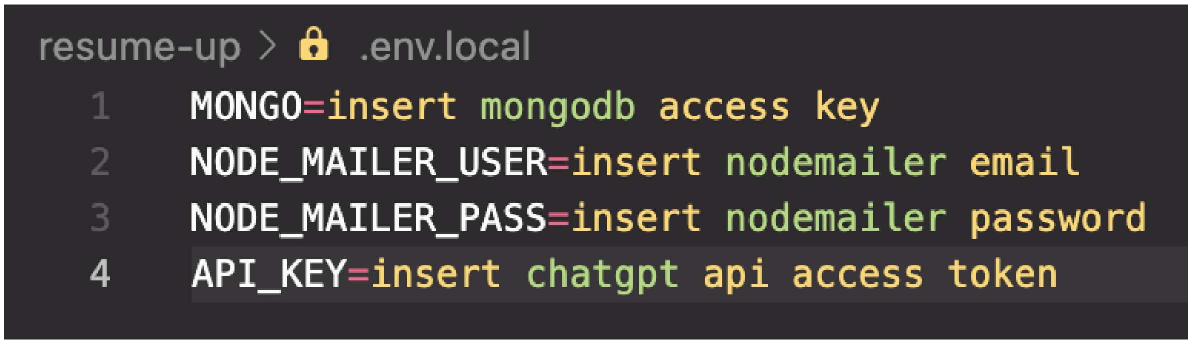
```
ResumeUp % cd resume-up
resume-up %
```

Then you will need to install all the packages needed to run ResumeUp. This can be done by entering the following command: "npm i":



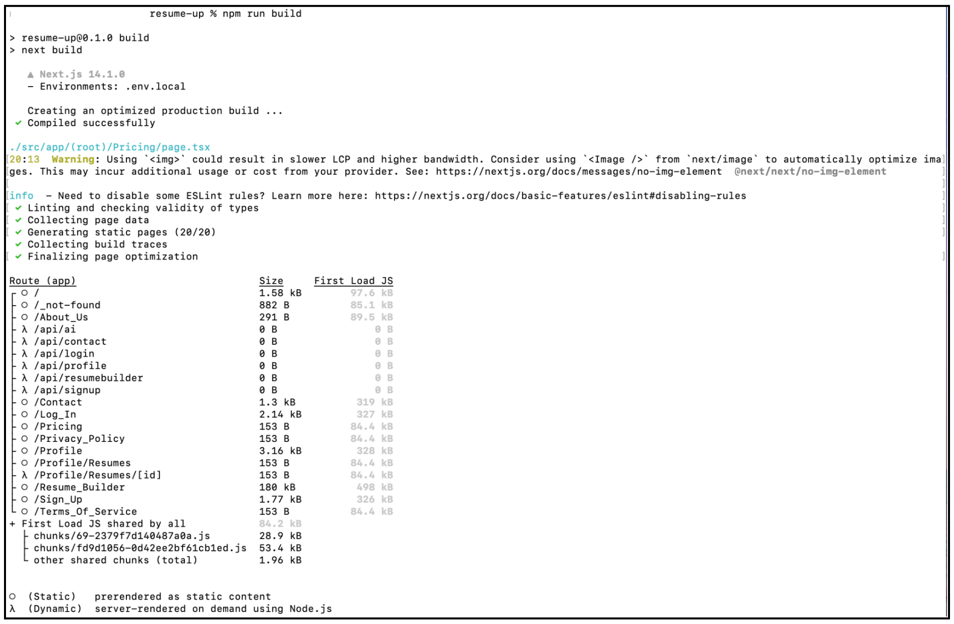
```
                        resume-up % npm i
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated @babel/plugin-proposal-object-rest-spread@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-object-rest-spread instead.
npm WARN deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.

added 1506 packages, and audited 1845 packages in 54s

214 packages are looking for funding
  run `npm fund` for details

3 critical severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

Since things like MongoDB access require a private key you will need to create a ".env.local" file inside of the "resume-up" directory. This can be done with a text editor of your choice. Inside the file, you will need to create a few environment variables that are used throughout the app. Those variables are MONGO, which is used for our database connection, NODE_MAILER_USER and NODE_MAILER_PASS, which are used for our contact form, and API_KEY which is used for our ChatGPT tokens. For these values, you will need to get your own access keys for each of the services, as we cannot share our private keys.



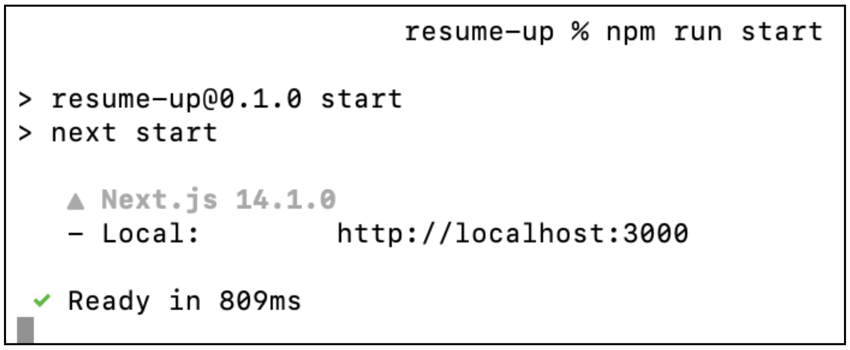
```
resume-up > .env.local
MONGO=insert mongodb access key
NODE_MAILER_USER=insert nodemailer email
NODE_MAILER_PASS=insert nodemailer password
API_KEY=insert chatgpt api access token
```

You will then need to build the project. This can be done by entering the following command: “npm run build”:

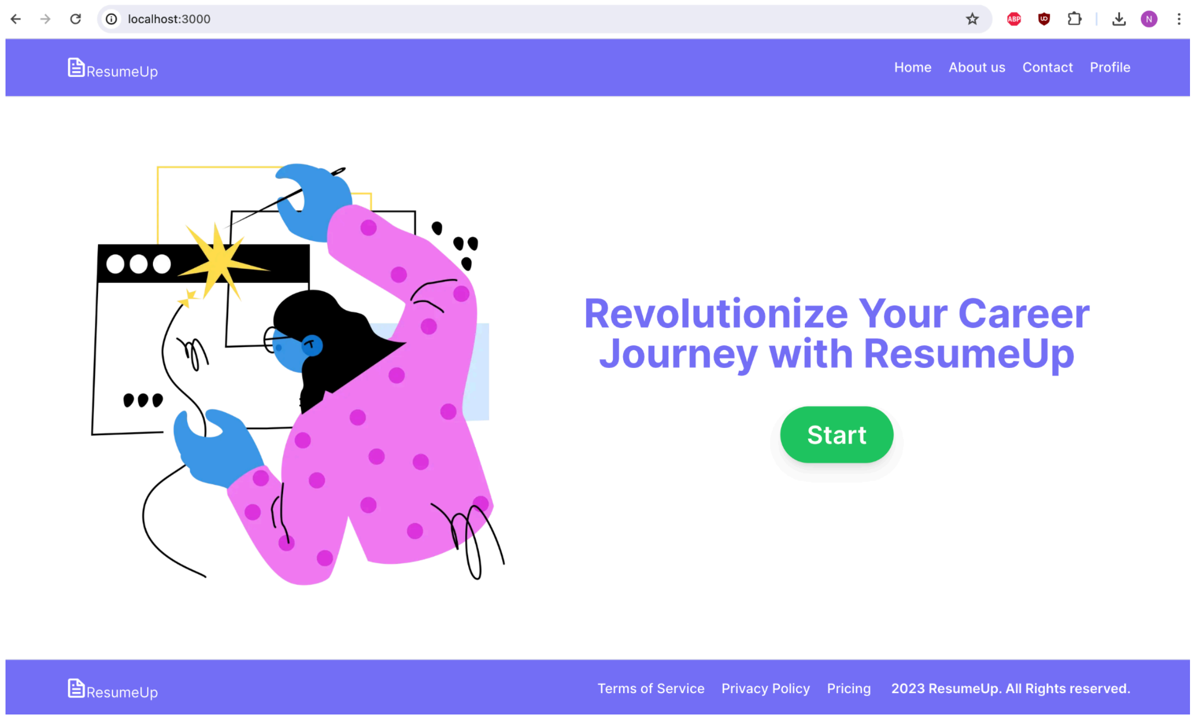


```
                       resume-up % npm run build

> resume-up@0.1.0 build
> next build

   ▲ Next.js 14.1.0
   - Environments: .env.local

   Creating an optimized production build ...
 ✓ Compiled successfully

./src/app/(root)/Pricing/page.tsx
20:13  Warning: Using `<img>` could result in slower LCP and higher bandwidth. Consider using `<Image />` from `next/image` to automatically optimize ima
ges. This may incur additional usage or cost from your provider. See: https://nextjs.org/docs/messages/no-img-element  @next/next/no-img-element

info  - Need to disable some ESLint rules? Learn more here: https://nextjs.org/docs/basic-features/eslint#disabling-rules
 ✓ Linting and checking validity of types
 ✓ Collecting page data
 ✓ Generating static pages (20/20)
 ✓ Collecting build traces
 ✓ Finalizing page optimization

Route (app)                              Size     First Load JS
┌ ○ /                                    1.58 kB        97.6 kB
├ ○ /_not-found                          882 B          85.1 kB
├ ○ /About_Us                            291 B          89.5 kB
├ λ /api/ai                              0 B                0 B
├ λ /api/contact                         0 B                0 B
├ λ /api/login                           0 B                0 B
├ λ /api/profile                         0 B                0 B
├ λ /api/resumebuilder                   0 B                0 B
├ λ /api/signup                          0 B                0 B
├ ○ /Contact                             1.3 kB          319 kB
├ ○ /Log_In                              2.14 kB         327 kB
├ ○ /Pricing                             153 B          84.4 kB
├ ○ /Privacy_Policy                      153 B          84.4 kB
├ ○ /Profile                             3.16 kB         328 kB
├ ○ /Profile/Resumes                     153 B          84.4 kB
├ λ /Profile/Resumes/[id]                153 B          84.4 kB
├ ○ /Resume_Builder                      180 kB          498 kB
├ ○ /Sign_Up                             1.77 kB         326 kB
└ ○ /Terms_Of_Service                    153 B          84.4 kB
+ First Load JS shared by all            84.2 kB
  ├ chunks/69-2379f7d140487a0a.js        28.9 kB
  ├ chunks/fd9d1056-0d42ee2bf61cb1ed.js  53.4 kB
  └ other shared chunks (total)          1.96 kB


○  (Static)   prerendered as static content
λ  (Dynamic)  server-rendered on demand using Node.js
```

Now, to run ResumeUp, you need to start the application. You can do this by running the following command: “npm run start”:



```
                                 resume-up % npm run start

> resume-up@0.1.0 start
> next start

   ▲ Next.js 14.1.0
   - Local:        http://localhost:3000

 ✓ Ready in 809ms
```



To use a live version of ResumeUp instead, you can head to this link: ResumeUp

7.2 Appendix 2 - Alternative Designs

We swapped from Amplify to EC2 and MongoDB. AWS dropped support for Next.JS 14 App Router and had little documentation to demonstrate how to run and set up the Amplify app so we decided it would be best to steer away from it. MongoDB, given our past experiences, allowed us to use our prior knowledge to set up our database quickly and with no cost to set up.

7.3 Appendix 3 - Other Considerations

You will not be able to run this project locally since we use our own private keys. Outside users will not be able to run the code properly without their own access keys. We also had to work with the department at Iowa State to get funding for this project. We learned new technologies like Next.Js and cloud tools.

7.4 Appendix 4 - Code

- ResumeUp Code
- Npm i to initialize the program. Since things like database access and ChatGPT tokens use secret keys, using just the code will not work the same for you. You will need to create your own .env file and include your own access keys.