

EECS 581 Project Proposal

Team No. 19

Nolan Blankenau, Theodore Harbinson, Harrison Heeb, Warren Wang, Henry Williams

Project Name:

In-browser IDE

Project Synopsis:

An in-browser IDE with Machine Learning features to allow developers to write and run code all from a web browser.

Project Description:

For a long time, the web browser was simply a way to view relatively simple websites or web apps. Nowadays, internet and web browsers are fast enough to let users run any app with seemingly no limit to their complexity. We see this with web apps such as 3D CAD, design software, and 3D games. With applications that used to only be possible as a pc or mac app moving to the web, cheaper computers with basic hardware are becoming more and more capable by just including a web browser. For lots of people, a Chromebook may be all they need. Our team wanted to extend this idea to software engineers as well. What if all your development could be done on a web browser? That is why we are building an in-browser IDE that can write and run code all on the web. Users will be able to go to our website, log in, and be presented with a text editor that will feel at home to them. They will be able to open a directory of a project and start editing code. At any point, a user can save their code to come back to it later, or to execute the code with the results printed to a console. Our team also wants to make a text editor that is better than average. In order to ensure that users can write well-documented code without hassle, we will be implementing auto comment generation using machine learning.

Project Milestones:

First Semester:

- 10/25/2020 - Complete research on tools we can use to accomplish our task and plan on how to efficiently use them.
- 11/13/2020 - Complete a port of an open source text editor for use on the web.
- 12/04/2020 - Complete the frontend (user interface) of our IDE with navigation built in.

Second Semester:

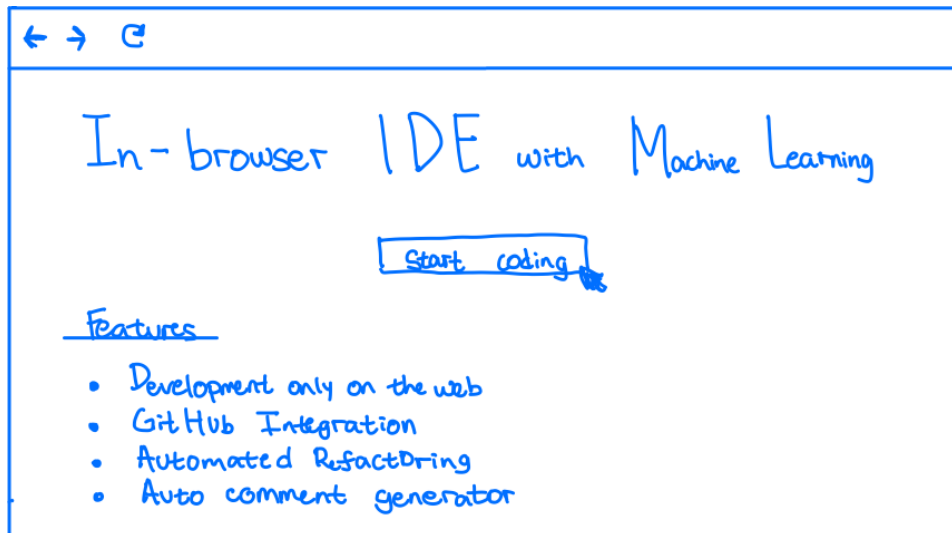
- 02/15/2020 - Complete ability to run code that a user writes and show the results on the frontend.
- 03/29/2020 - Complete ability for a user to save their files and folders to our IDE.
- 04/10/2020 - Complete auto comment generation of user written code.
- 05/03/2020 - Final product ready.

Project Budget:

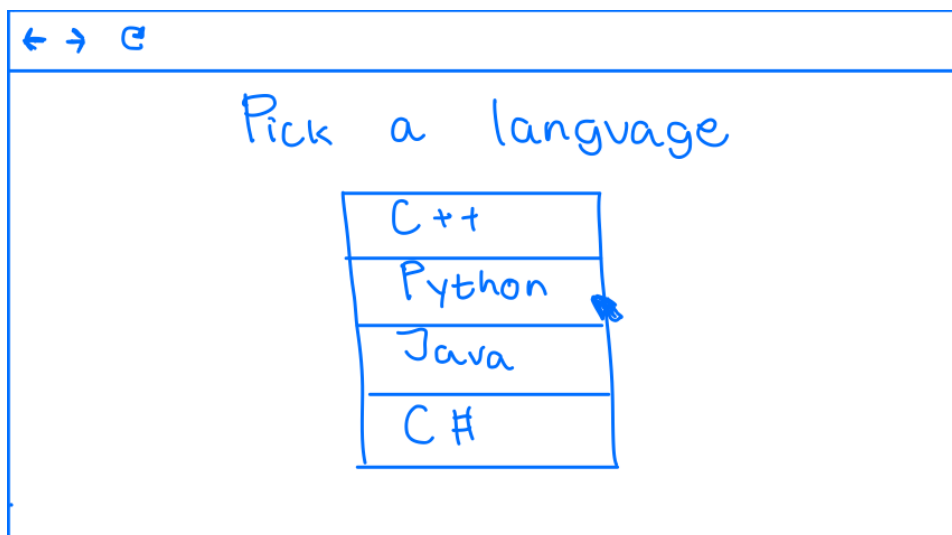
\$0 – We plan on hosting our code with Heroku’s free hosting service and use only free open source software.

How the Software Works + Illustrations (Rough Product Design):

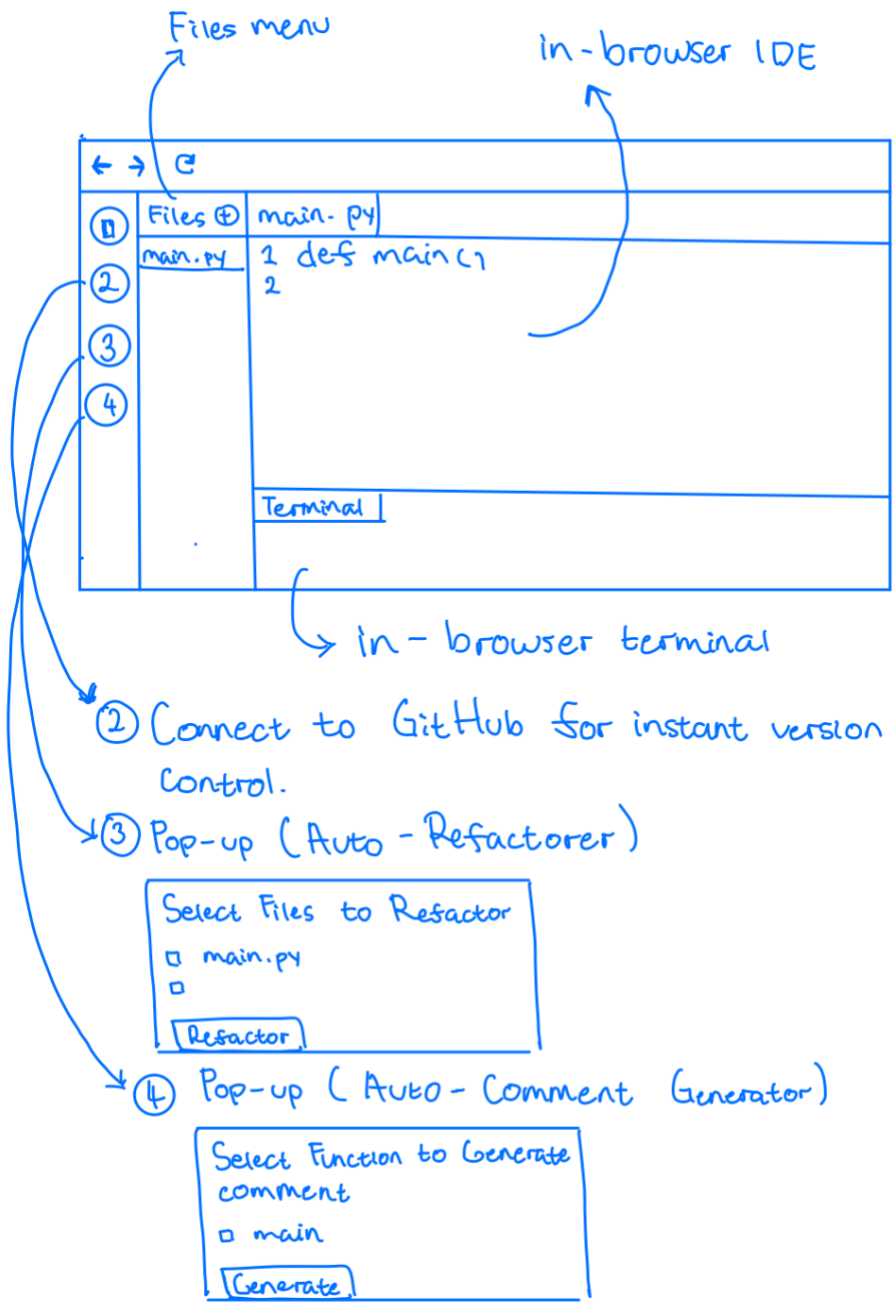
NOTE: The cursor located on each illustration represents what button a user will press to move forward with the product.



When the user enters the website address in the browser, they will be led into the landing page above. The pages will all be structured using HTML and styled using CSS. We will also use JavaScript to add logic to the buttons. The landing page contains information regarding our product features and a button that states “Start Coding” that will route us to the second page below after pressing it.



The second page allows the user to pick a programming language which the user wants to code in on the in-browser IDE. We plan on starting with one programming language for our product and slowly add more based on feedback and demand. Once the user clicks the language of their choice it will take that information and send a request to the Backend server (Node.js or Flask) which will then route the website into the matched programming language IDE and terminal.



We reached the main component of our product. The main portion of the page consists of the text editor which the user will code in using their desired language. Below that is the terminal which the user can use to run their program. Over to the side of that is the files column. On the file's column, user will be able to create, rename, and delete files for their program. Every time a user creates, deletes, or renames a file, we will update it on our database (PostgreSQL) in real-time using the Node.js/Flask server to send that request. Finally, the left-most column contains access to our version control and Machine Learning automated comment generator and refactorer feature. When we click the number 2 button above, it will allow us to sign into our GitHub account and connect to the desired repository. When we click the number 4 button above, it will prompt a pop-up on the website. The pop-up will consist of all the functions with its corresponding file our algorithm has detected on the program with checkboxes on the side of it. The user will then be able to check mark the functions it wants to generate the comment for and after clicking generate, our Machine Learning model (ML Library) will try to output the comment describing the function and code. Regarding button 3, it will also work the same but instead of generating code, it will refactor the code. For our capstone we plan on excluding button 3 unless we have the time to effectively build it.

Design Constraints:

Technical Constraints:

- JavaScript (Frontend)
- Node.js/Flask (Backend)
- PostgreSQL (Database)
- Must be fast
- Must utilize server-side processing
- Needs to have flexibility and run on multiple users at once
- Needs to utilize open source code
- Requires machine learning libraries

For this project, our team will need to implement an application that is both quick and easy to use. Because of this, the project will have many technical constraints. The most important constraint for the project is the bandwidth for the user to be able to edit code at real-time and get output from their programs as soon as it had been delivered. The best technique, and the technique that more and more companies are using for their applications, is to use server-side processing. By offloading the main processing to a server, the user's computer will have to do less work and provide the user with a better experience. This requirement will limit the pool of libraries that we can develop with for the project. It will need to be easy to develop with and offload processing to a server. Frameworks like NodeJS and Flask are good for this. Our application will also need to be very flexible so that multiple users can run their program at once. This means that there will need to be a variable number of channels as more and more users are on the website. Each user needs to have their own server space and channels of communication. The intent of this application is that each user on the site has their own set of files to work with and their own space to compile their code. This will require a flexibility in the amount of

compute power that is being used, similar to the way that AWS resources can be configured. The project is going to need to utilize open source frameworks as well. Because we do not have the time or money to spend on great libraries or develop a massive project, it is essential that we are able to use open source or free code. Finally, a portion of this project utilized machine learning to generate comments on code. This task can be completed with the right machine learning libraries and the right data. We will need to be able to collect enough data on code and comments to make a model that works well. This can be achieved using open source code on GitHub and the endless resources that can be found online. The model that we create for generating comments can be done using TensorFlow, as well as other natural language processing libraries online.

Business Constraints:

- Must be usable in a browser
- Must be completed by the end of spring semester
- Project must operate within our budget
- Accessible to any laptop with a browser

The main purpose of this project is to provide developers with a way to code purely in-browser. This business constraint means that everything we do must be accessible from only a browser. This will affect the amount of development that we do with different languages. We will need to implement a UI framework to make the user experience as good as possible. This project also has a time limit. We must be finished with the entire project by the end of spring semester 2020. That means that we have less than 6 months left to work on this project. We will need to organize our work into sprints and keep our progress on track every two weeks in order to meet that deadline. This project also has a budget limit. Our project should not cost a lot in theory, considering that we will not start with a large user base. If we were going to sell this product to a lot of people, we would have to spend a lot of money on cloud computing resources like AWS. Our last business constraint is that the product must be usable across all laptops, no matter the processing power or size. In order to do this, we must use UI frameworks and create an architecture that is usable across all devices and requires a low amount of processing power to run. This should be simple, considering that most of our UI will be text based.

Ethical Issues

Most ethical issues that will arise from this project will be privacy related. Since we will be storing code and information from users, we need to make sure we don't breach ACM Code of Ethics 1.6, 1.7 and 2.9. Usually user data on a server could just be encrypted to keep information private; however, since we are storing code that needs to be run on the server, encryption would not be possible in this scenario. Since we cannot properly encrypt data on the server, we cannot store any data on our system and keep the user's privacy a priority. Therefore, we require that code storage be done off the server. This could be done by hosting on GitHub or storing on the user's computer. This would also decrease the amount of required storage for our server, reducing our budget if expanding the project. We also need to keep codes 1.2 and 2.9 in check

since we are running foreign code on a server. This makes things very complicated and messy, as it will require setting up virtual environments for every user and making sure that every user doesn't have access to system resources, information, or other users' information or data. All of this needs to be kept in mind while we work on this project, as not finding a solution to any of these issues would make our product borderline unusable and dangerous to the public.

Intellectual Property Issues

We are certainly not the first ones to come up with a web-based IDE. The most popular browser-based coding website is repl.it, which allows the user to write and run code in 50+ languages, including some esoteric languages. There are many others for specific uses, such as Amazon's Cloud9 IDE which comes fully integrated with AWS and CodeAcademy which is for simply learning to code. We don't aim to copy or compete with websites like these; we just want to provide another option with our own unique takes on some features. Because there are so many projects like this, and the idea of an IDE is so broad, we don't imagine there will be any intellectual property issues.

Change Log

Our team originally wanted to build a script that would automatically do a complete refactoring of a codebase. Upon realizing this would be difficult to split up word loads for, we pivoted to our current idea.

We had thought that we would build a text editor from scratch, however, now we are planning on porting an existing open source text editor to work on the web.