

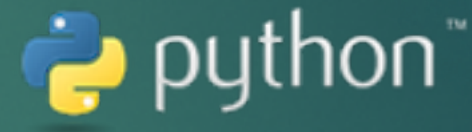
FLOWCODE

BY ARCADIA

Problem

- ▶ Current programming methods are abstract and hard to learn.
- ▶ There is not a solid bridge between programming and its end products. Thus, developers and customers are not able to make healthy discussions about their needs and capabilities.
- ▶ We aim to redirect comprehensive load from people's brains to processors by visualising the functionality, flow and structure of source codes.

Used Technologies



django

Product Features

- ▶ It is a web based application.
- ▶ We support well-known web browsers such as Mozilla Firefox, Google Chrome, Internet Explorer and Safari.
- ▶ No additional software installation is required. All necessary software is installed in our server side.
- ▶ Our target audience is those who have just started learning programming.

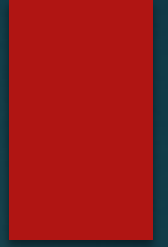
Product Features

- ▶ The users can sign up or, if they already have an account, sign in.
- ▶ For sign up, user name, password and e-mail is required.
- ▶ All users have their own working space.
- ▶ The users can freely add new files or work on their previously created files.
- ▶ Renaming the files is available.
- ▶ Main parts of the product is : Front-End, Back-End, Debugger and Data Structure.

Front-End Development

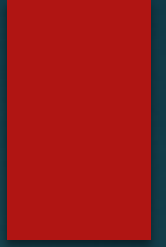
- ▶ The user has two options in the front-end:
 - ▶ Write / Upload a source code
 - ▶ Draw a Flowchart
- ▶ In the first option, the source code is sent to server side and compiled. If it gives an error, the error message is shown on the browser side. Otherwise, that source code is parsed on the server side.

Front-End Development



- ▶ The communication between server side and client side is done via JSON objects. These JSON objects are traversed recursively to find out and draw each element.
- ▶ To visualize the parsed code on the client side, we used the JointJs library which is a JavaScript library to create interactive diagrams and visualize graphs. We created generic shapes for our purposes such as shape for condition statement and shape with additional menu (with delete, change, add link and add breakpoint features) for all statements.

Front-End Development



- ▶ For the second option, we created an element menu which includes variable declaration and assignment, if statement, loop statement and function calls. These elements can be added to the FlowChart canvas easily and later on modified if necessary.
- ▶ On addition to the FlowChart canvas, we check to which element the selected element is wanted to linked. Accordingly, we arrange the color of the selected element.

Front-End Development Elements

- ▶ Each element has a specific color and shape so that it can be separated from others.

Name	Color	Shape
Variable Declaration & Assignment	Light Yellow	Rectangle
If Statement	Light Yellow	Diamond
Loop Statement	Blue	Diamond
Function Calls	Turquoise	Rectangle

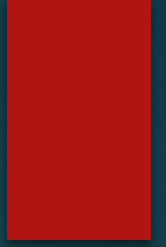
Back-End Development

- ▶ To communicate between client and server JSON's created by Python Modules which are built by our team are used and sent to client through Django Framework.
- ▶ Users' data (files, personal information) is also stored as encrypted in MySQL database using helper functions of Django Framework to keep them unreachable and unreadable. To protect files to be reached by unauthorized user, we are controlling user permissions for each file before reaching database of files.

Back-End Development

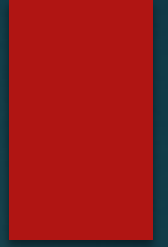
- ▶ In Code to Flowchart conversion, we are using abstract syntax tree (AST) to create corresponding JSON representation of the code written by user. AST is a unique representation of the code which shows abstract syntactic structure of source code.
- ▶ To create AST from the source code, we used Clang compiler. Clang takes filename as an input and generates AST from that file. And after getting AST, we have generated corresponding JSON.

Back-End Development



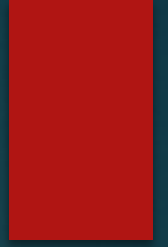
- ▶ To decrement data size, and keep client from unnecessary information, while we are traversing AST, we have just kept certain information in JSON of source code.
- ▶ Clang supply this feature for C,C++ and objective C. Thus, we parse C and C++ yet.
- ▶ In Flowchart to Code conversion, we have generated unique filename for each session and generated code using JSON that is produced from the client.

Debugger



- ▶ In addition to the ordinary debugging on the code editor, we offer visual debugging which is performed on the flowchart. The user is able to see the flow of his program, pause it, go step by step or terminate it.
- ▶ Moreover, variable values can be observed. The user can add a watch to any variable he wants and inspect its value while debugging.
- ▶ It is also possible set breakpoints on both flowchart and code editor.

Data Structures



- ▶ Visualize the static and dynamic data structures to have an overview of your system.
- ▶ For static, structs, unions and classes can be parsed and shown. On the other hand, for dynamic, in addition to the previously mentioned structures, linked lists and arrays can be visualized.



- ▶ Further information can be found in the website of our product:

www.flowcode.tk