The Developer Environment

Intro to Developer Tools

As a Software Engineer, you'll come to rely on tons of tools, applications, and plugins in order to work efficiently. These tools, applications, and plugins come together to form your developer environment. Getting used to all of the parts of your environment can be an overwhelming task, and determining which of these tools, applications, and plugins to use is a huge task within itself!
Let's check out some of the environments you may encounter while on your Software Engineering journey.

Integrated Developer Environment

Probably the most popular solution, the Integrated Developer Environment is an all-in-one text-editor that includes all of the resources you need to write, edit, organize, test, and deploy code. IDEs are often dedicated to a specific programming language or framework.

Resource

Codecademy has a great writeup on what an IDE is. Check it out here

Pure Text Editor

For a little more of a fragmented approach, using a Text Editor is a great option. Text editors range in capability. Some include cool color schemes, code completion, and automatic code formatting whereas others are more lightweight and dedicated to providing you with a space to code.

Once you have your text editor up and running, you can use other tools to do any tasks you require, whether that's building/compiling code, managing version control, or unit testing.

Resource

Kinsta has compiled a list of workflow-enhancing Text Editors.

100% Terminal

For a more barebones approach to coding, there's always the terminal. For example, Vim is a command-line interface that allows you to write code directly in a terminal with the help of keystrokes, commands, and scripts.

Resource

Check out this funny DEV.to post about stumbling upon vim and ultimately falling in love with it!

Setting Up Your Developer Environment

When setting up your developer environment, the tech stack you plan to use can help you determine which tools you might need.

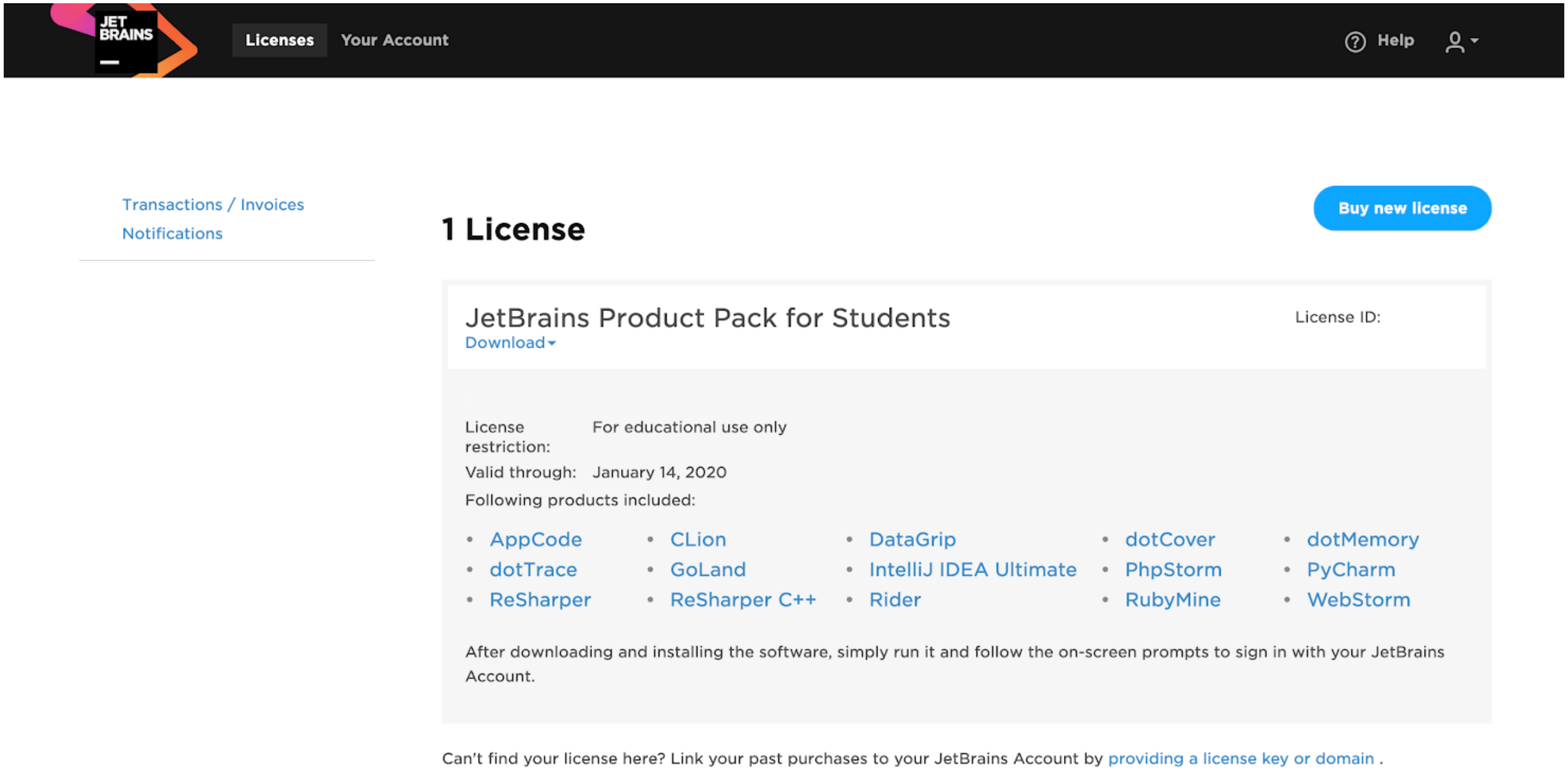
A tech stack is the list of the programming language(s), tools, and frameworks your project requires. You don't have to understand how the tech in your project's tech stack works, nor do you have to master it. Just knowing what each item in your tech stack will be used for in your project is a great start!

Here at the MoraL Code, we lean towards the IDEs made by a company called JetBrains, who are known for their Java IDE IntelliJ. As a college student, you can get the professional version of any JetBrains IDE for free by signing up with your university (someone@school.edu) email! Here’s how to do it.

1. Sign Up at [JetBrains Products for Learning](https://www.jetbrains.com/).

2. Log in to your student account at https://account.jetbrains.com/login

3. Once logged in, you’ll see a list of all the IDEs you have access to.



1 License

JetBrains Product Pack for Students

License ID:

License restriction: For educational use only

Valid through: January 14, 2020

Following products included:

- AppCode
- CLion
- DataGrip
- dotCover
- dotMemory
- dotTrace
- GoLand
- IntelliJ IDEA Ultimate
- PhpStorm
- PyCharm
- ReSharper
- ReSharper C++
- Rider
- RubyMine
- WebStorm

After downloading and installing the software, simply run it and follow the on-screen prompts to sign in with your JetBrains Account.

Can’t find your license here? Link your past purchases to your JetBrains Account by providing a license key or domain .

   Click on the JetBrains IDE appropriate for the language you’d like to code in then download and install.

4. Open your IDE and sign in with your JetBrains student account!