PORTFOLIO

Engineering—Computer Programming

Yubo Cao

Chemistry Language

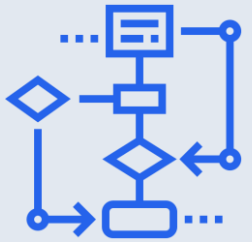
Website & Blog

FTC Robotics

Jqboard

Redo List

Quizlet Helper

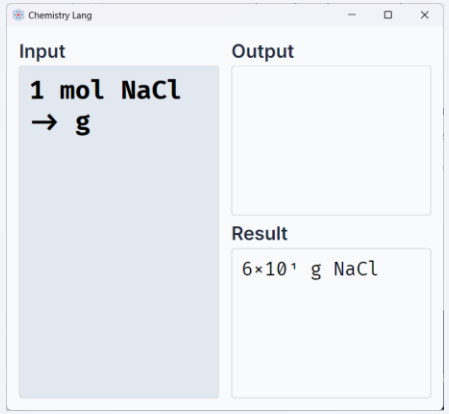


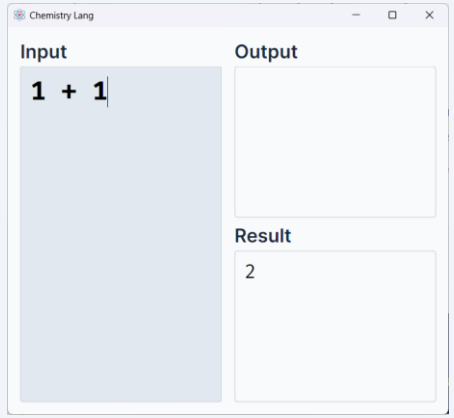
Algorithms

Better Gradebook

Chemistry Language

A calculator that can solve chemical questions efficiently. However, it goes far beyond a calculator—it is a functional, expression-minded programming language with keywords that are adjusted to represent the student's daily life with sweet syntax.





$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{1n} & \dots & a_{nn} \end{bmatrix}$$

Linear Algebra

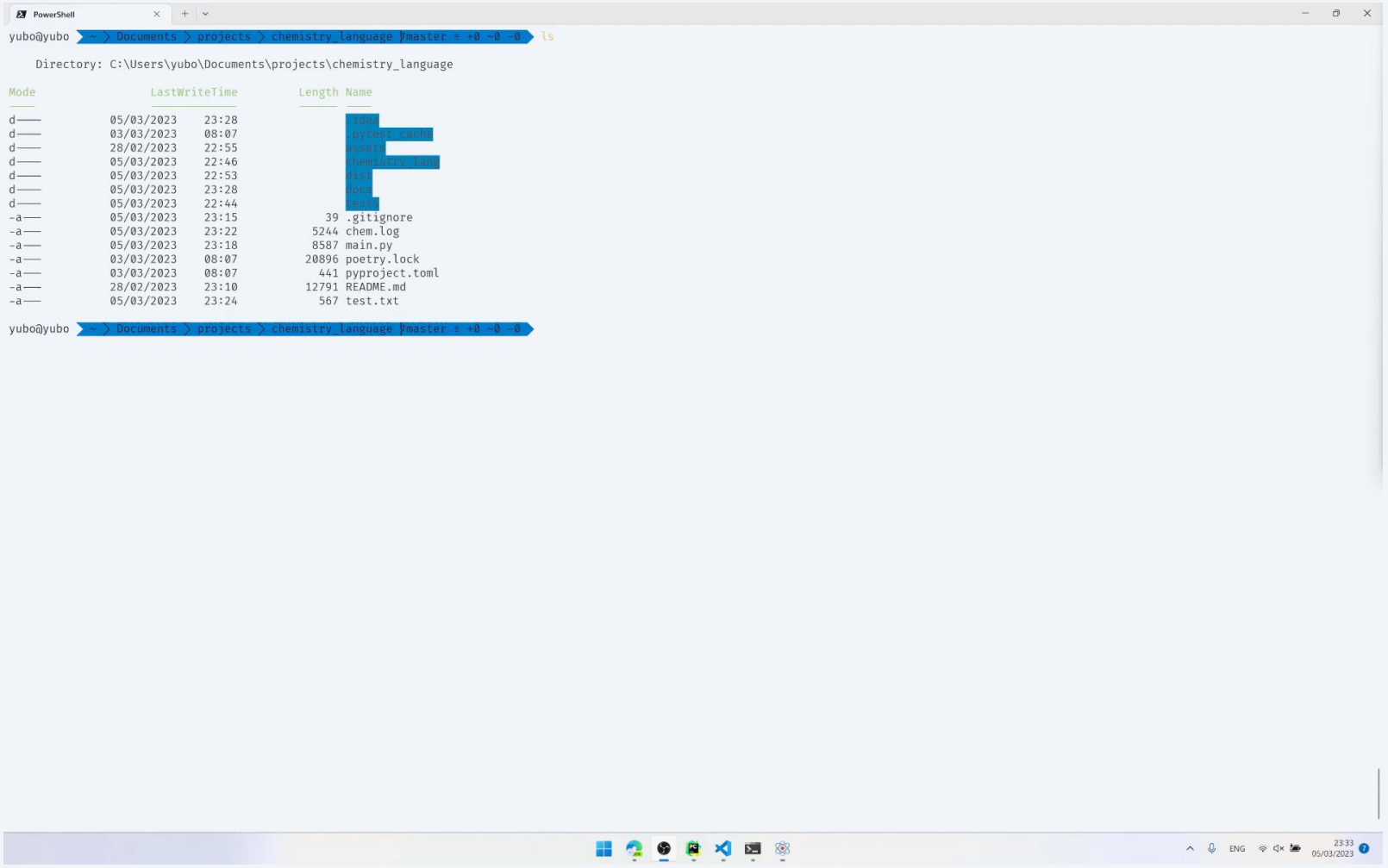
Abstract Syntax Tree

Python

Recursive Descent Parser

Context Free Grammar

Chemistry Language

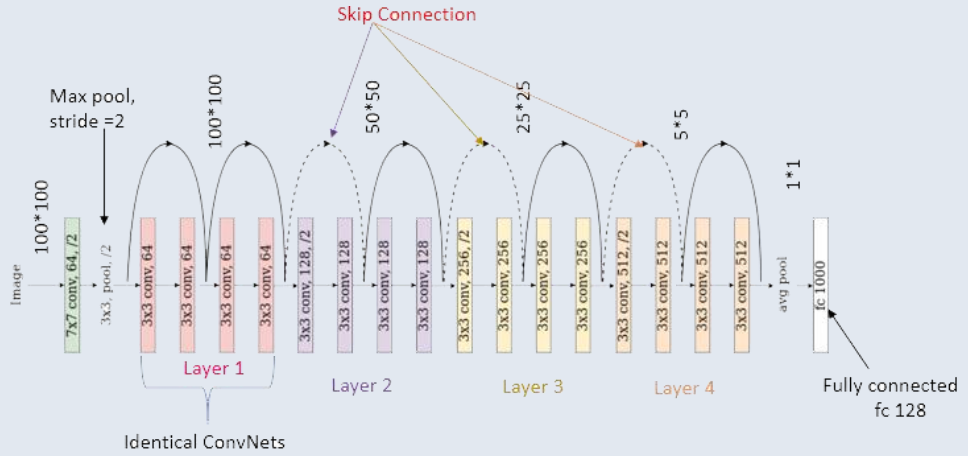


FTC Robotics

This project involved training a deep learning model using PyTorch and the Colab platform. The project aimed to develop a model that could accurately detect and classify objects in images captured by a robot.

To achieve this goal, I collected 600 images from the robot and used Label Studio to label them.

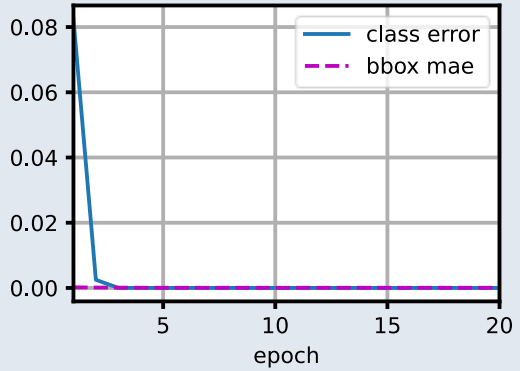
Single shot multibox detector with ResNet18 as the backbone. $\mathrm{b} \times 256 \times 256 \times 3$ input are accepted, and generated $\mathrm{b} \times n \times 4$ predicted offsets.



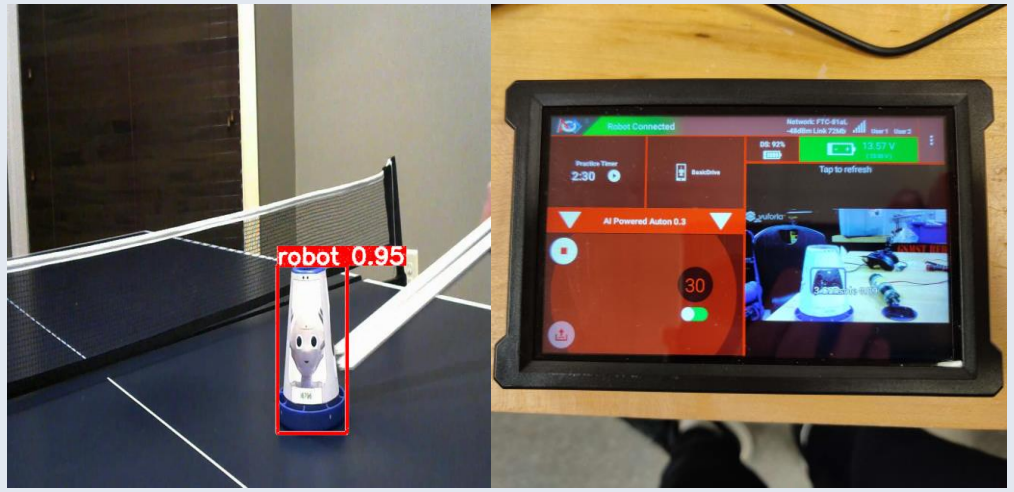
ResNet-18 Architecture

FTC Robotics

The model performs classification using receptive fields and employs various techniques such as weight decay, alternative learning rate, and transfer learning in order to be trained successfully and achieve good accuracy.



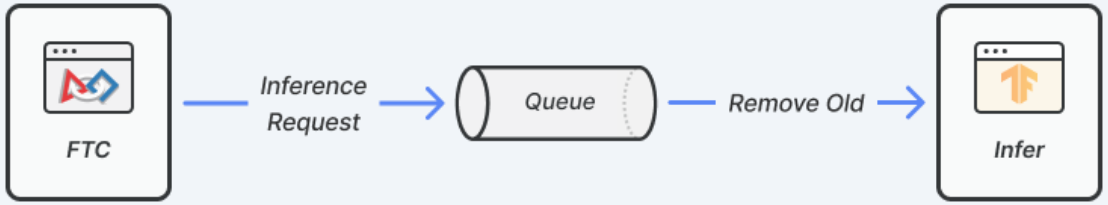
A post-processing layer is added to perform Non-Maximum Suppression (NMS). Finally, the model is exported to the ONNX runtime and converted to a tflite file for deployment on an Android-based robot.



FTC Robotics

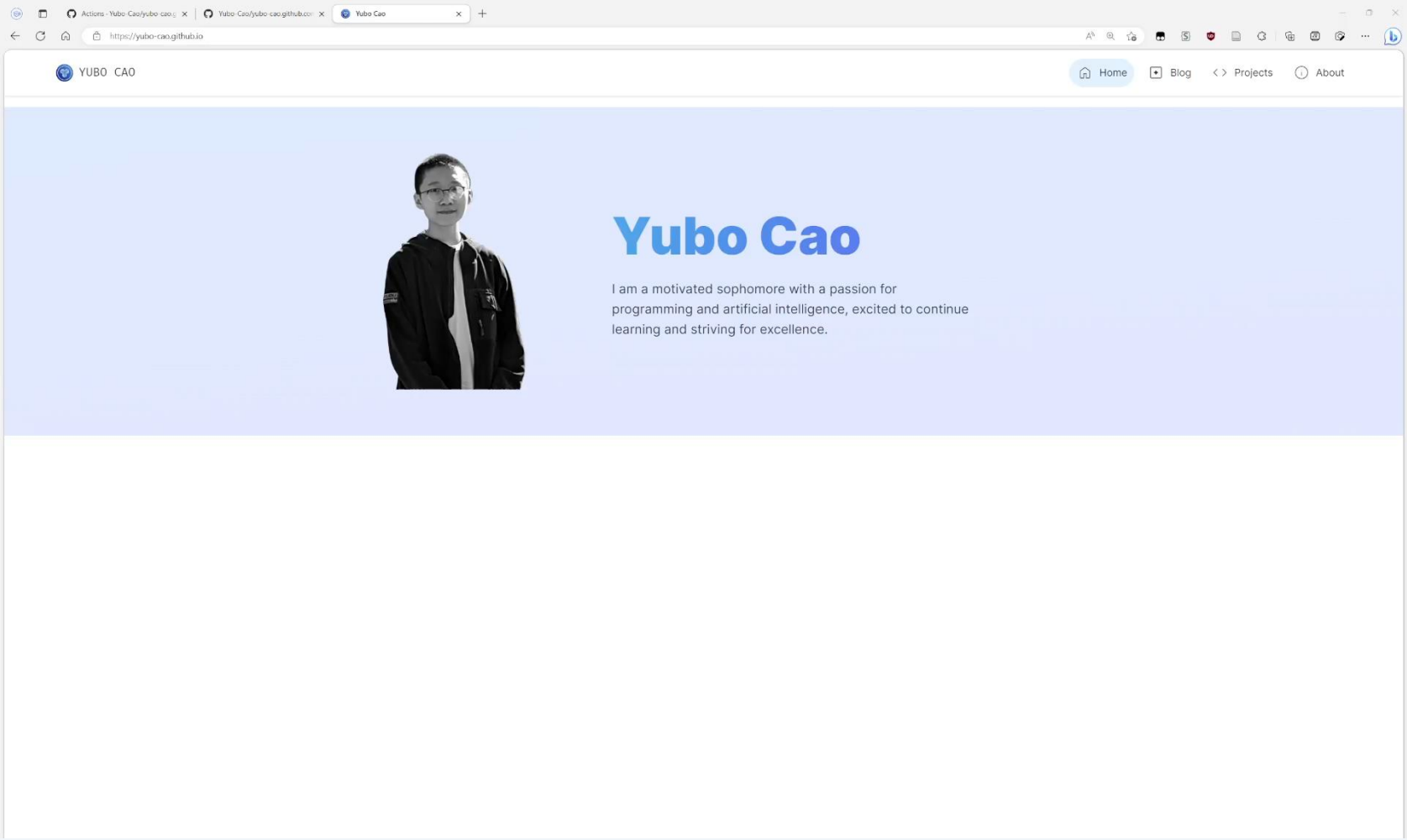
To improve performance, I have implemented multi-threading to perform operations in parallel; furthermore, requests for model inference are stored in a queue, allowing for efficient processing and response times.

I have also developed an object-oriented system for drive train control and movement. This system allows us to control the movement and positioning of our robot in a more precise and efficient manner.




```
override fun processFrame(input: Mat): Mat {
    val width = input.cols()
    val height = input.rows()
    preprocess(input)
    val thread = Thread {
        model.run(inputArray, outputArray)
    }
    thread.start()
    thread.join(TimeUnit.SECONDS.toMillis(5))
    _detections = outputArray[0]
        .filter { it[4] > minConfidence }
        .map { Detection.from(it, width.toDouble(), height.toDouble()) }
        .toMutableList()
    _detections = nms(_detections)

    for (det in detections) {
        Imgproc.rectangle(input, det.bbox.toRect(), COLOR, 2)
        val size = Imgproc.getTextSize(
            det.label,
            Imgproc.FONT_HERSHEY_SIMPLEX,
            1.0, 2, null
        )
        Imgproc.rectangle(
            input,
            Point(det.bbox.x, det.bbox.y),
            Point(det.bbox.x + size.width, det.bbox.y - size.height),
            COLOR,
            -1
        )
    }
    return input
}
```

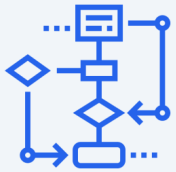
Personal Website



Quizlet Helper

The Quizlet Helper project offers a technical solution to the tedious task of manually managing folders and study sets in Quizlet. Through the use of a web driver (Playwright), the project provides an object-oriented API that automates the creation and management of these elements. This is accomplished using properties and descriptors, resulting in a Pythonic and intuitive interface. Additionally, the API is idempotent, ensuring that repeated calls to create the same elements will not result in duplications.

```python
with open("auth.yml", "r") as f:
    config = load(f, Loader=Loader)
    password, username = config["password"], config["username"]

with sync_playwright() as p:
    browser = p.chromium.launch(headless=False)
    user = User(username, password, browser)
    folder = Folder(user, name="Barron")
    folder.created = True
    set = StudySet(
        user,
        name="Barron 1",
        cards=[
            Card("abandon", "放弃"),
            Card("ability", "能力"),
        ],
        definition_lang="中文（简体）",
        word_lang="英语",
    )
    set.created = True
```



Algorithms

The algorithm repository, co-developed by myself and my teammate Anish, contains a collection of algorithm problems that we have successfully solved. These problems are accompanied by detailed notes, designed to aid in the learning process for newcomers.

```cpp
void dfs(int u) {
    if (u == n) {
        for (int i = 0; i < n; i++) cout << path[i] << " ";
        cout << endl;
    }
    for (int i = 0; i < n; i++)
        if (!used[i]) {
            path[u] = i + 1;
            used[i] = true;
            dfs(u + 1);
            used[i] = false;
        }
}
```

Discretize

- Discretization is a technique that maps a large range of numbers to a small range of numbers. For example, if we have numbers from `0` to `1e9`, we can map them to `0` to `1e5`, since it is impossible to create an array of size `1e9`.
- There are some caveats:
  - There might be duplicate numbers, so we need deduplication.
    - First sort the array, `sort(a.begin(), a.end())`.
    - Then `a.erase(unique(a.begin(), a.end()), a.end())` will remove all the duplicate numbers in the array `a`. `unique(a.begin(), a.end())` will move all the unique numbers to the front of the array, and return the pointer to the first duplicate number. `a.erase()` will remove all the duplicate numbers.
  - How to properly map the numbers to the new range.
    - A binary search can be used to find the index of the number in the new range. Find the first position that is larger than or equal to the target in the deduplicated, sorted array.

```cpp
int main() {
    vector<int> a;
    int n;
    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i];
    sort(a.begin(), a.end());
    a.erase(unique(a.begin(), a.end()), a.end());
}

// find the first position that is larger than or equal to the target
int discretize(vector<int> a, int x) {
    int l = 0, r = a.size() - 1;
    while (l < r) {
        int mid = l + r >> 1;
        if (a[mid] >= x) r = mid;
        else l = mid + 1;
    }
    return r + 1; // depends on the problem
}
```

Sum in Interval

Jqboard

jqboard is a cross-platform library designed to handle clipboard functionality.

The library provides support for copying and pasting the following formats:

- Image/png (PIL.Image.Image)
- Text/plain (str)
- Text/html (lxml.etree.ElementBase)

With its Pythonic design, jqboard is simple and straightforward to use.

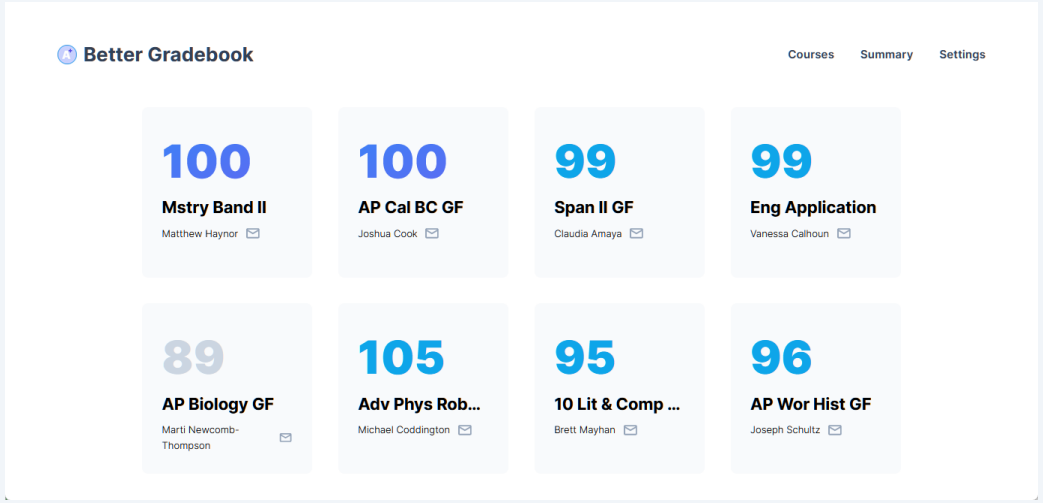
Internally, meta-programming is used to centralize dispatch in a single place, rather than spread if else statements everywhere.

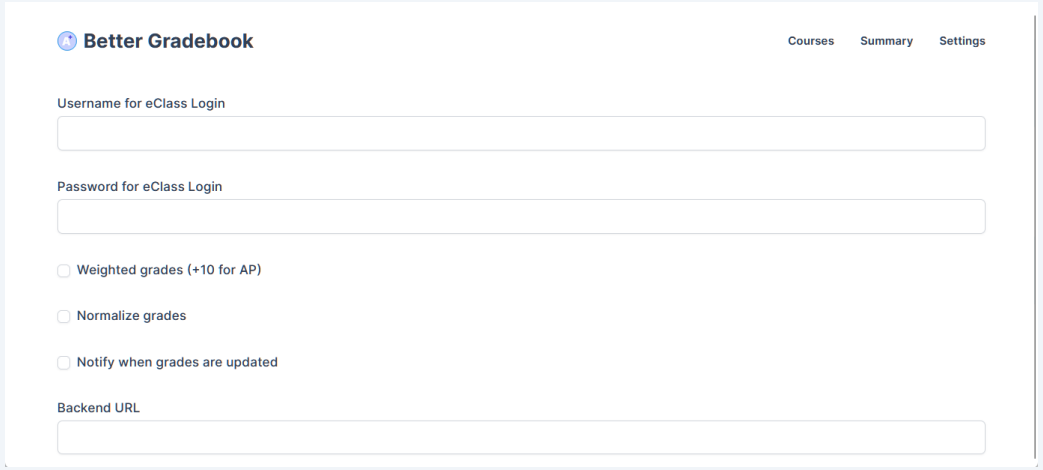
```python
clip = Clipboard()
print(clip.paste(ClipboardFormat.TEXT))  # HTML, IMAGE

try:
    clip.copy("Hello World") # smart format detection
    clip.copy("<h1>Hello World</h1>")
    clip.copy(Image.open("tests/assets/picture.png"))
except ClipboardError as e:
    print(e)
```

```python
def __new__(cls, *args, **kwargs) -> "Clipboard":
    if cls is Clipboard:
        plat = Platform.current()
        if plat == Platform.WINDOWS:
            from jqboard.win_clipboard import WindowsClipboard

            cls = WindowsClipboard
        elif plat == Platform.LINUX:
            from jqboard.linux_clipboard import LinuxClipboard

            cls = LinuxClipboard
        else:
            raise NotImplementedError("Unsupported platform")
    return super().__new__(cls)
```

Better Gradebook

This project offers a user-friendly interface for grade checking and notifications in an aesthetically pleasing manner. It features an asynchronous Python spider for backend functionality that fetches the grade, stores it in a cache, invalidates the cache periodically, and saves the historical grade in a SQLite database. Aiohttp, lxml, and other similar technologies are utilized for this purpose. The frontend is designed as another website and is powered by Next.js.





THANKS

Engineering—Computer Programming

Yubo Cao