**CodeX Learner**

**Specification Document**

**Chander Thakur**

**Last updated: September 19, 2023.**

**Team name: Git Guardians**

Note\* This Document is far from complete and iterations are being made, keeping in mind the team’s capabilities and the time limit.

**Overview**

**CodeX Learner** is a programming focussed e-learning platform created for user of any age and background to get started on the journey of programming from the very basics.

But it won’t just be for beginners, users of any level can learn their desired course from this platform, they can watch embedded YouTube videos, and read AI generated simplified content with examples for better understanding and an integrated Code editor to test their skills as they learn.

Examples of a few courses are Introduction to Computer Science, Basics of Python and Basics of C.

**Requirement analysis**

**Where and how to start?**

There are many people who have an interest in coding but cannot find a place to start. People who already have a career in another field but decide to take an interest in Programming. But they find the online courses available hard to follow or very expensive.

We want to build an e learning web app that can provide learners a way to start from the beginner level and follow a roadmap to go from beginner to advance in whatever they want to learn about programming.

**I don’t get it;**

Many popular sites have great content for learners but while they explain something they might use words that have a whole different meaning and now we have to open another tab to look up what word means, and this goes on.

Short and simple AI generated content with real life examples is the way we solve this problem.

**Here is how a popular site on web explains Functions in python :**

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

Creating a Function

In Python a function is defined using the *def* keyword:

Example:

*def myfunction():*

*print("Hello from a function")*

A beginner trying to understand functions might find this difficult to understand

**And here is how an AI explains functions:**

Think of a function in Python like a recipe in a cookbook. Just like a recipe tells you how to make a dish step by step, a Python function is like a set of instructions that the computer can follow to perform a specific task.

**Parts of a Python Function:**

1. **Name**: Every function has a name, just like every recipe has a name. For example, you can have a function called "MakeSandwich."
2. **Inputs (Parameters)**: Functions can take inputs, which are like ingredients for a recipe. These inputs are called parameters. For our "MakeSandwich" function, the parameters could be "bread," "cheese," and "lettuce."
3. **Instructions**: Inside the function, you write a series of instructions, just like the steps in a recipe. For "MakeSandwich," the instructions would be "Take two slices of bread," "Put cheese on one slice," and "Add lettuce on top."
4. **Output**: After following the instructions, a function can produce an output. In our case, the output of the "MakeSandwich" function would be a delicious sandwich!

**How to Create a Function:**

Let's create a simple Python function called "AddNumbers" using real-life examples:

*def AddNumbers(num1, num2): result = num1 + num2 return result*

Here's how it works:

* ***def***: This keyword tells Python that we're defining a function.
* ***AddNumbers***: This is the name of our function, like the name of a recipe.
* ***(num1, num2)****:* These are the parameters, like the ingredients you need.
* ***result = num1 + num2***: This line does the math, just like a recipe step.
* ***return result***: Finally, we give back the result, just like serving the finished dish.

**What are the success metrics?**

Does it work?

**Functional Specifications**

**Page 1(Login Page)**

The First thing a user will see when they open CodeX Learner will be a login page

It will have the following fields to be filled by user:

Email

Password

A (remember me) Checkbox to save user info to the browser

And the following buttons:

Login

Register

Forgot password

Home

About

Service

Contact

**{NON GOALS}**

The user will only need to sign up or login to continue and save their progress.

Since we are not acquiring any personal data like Name, Phone number, Gender etc. any user can use the platform without signing up, or logging in.

We have no intentions of Implementing security protocols anytime soon.

Forgot Password?

If the user forgets their password we will simply send a password reset form to their email.

After the first few iterations of the spec were done our fronted dev already started working on creating a possible Login page (open to variations).



**Page 2(Homepage)**

Next Page which is also the Homepage will have Things;

A dashboard of the user’s current progress of courses and quizzes so they may continue any course

-If the user did not Login this will just be a panel of the courses available

The dashboard will also have two buttons to allow users to continue reading their course content or attempt Quizzes related to the most recent chapter they completed.

A Search Button to search for a specific course (location of a button will be decided later).

-The search button will just filter out the courses according to keywords typed by user and show the results on homepage itself

**Page 3(Content Page)**

This is the on-click event page for when the user clicks on a course

This will be an interactive page with:

Details about the chapters

Content of each chapter

Quiz at the end of each chapter

A code editor

**Code Editor**

-Students can use an inbuilt code editor to practice code right next to the content

-API running on the backend server, which will take a piece of code and language as input and output the answer after running the code on the server

-Frontend code editor, we can choose the language and edit and modify the code here. Then we make a post request to the backend API and show output on the web Application.

-A Top Panel will consist of brief details about the course and a progress bar

-Embed videos from various sources on YouTube (after taking permission from owner).

-Integrate Categorised and personalised content taken from several open sources websites.

-Get permission from the creator and embed YouTube videos, the right video for the right topic.

-Add Captions and transcripts for videos.

-Integrate problems similar to one’s found in Leetcode and Hackerrank.

**Future Scope;**

Another interface for instructors

Who will be the instructors: YouTube Content Creators

We have seen that many teachers on YouTube use the comment section or the description to put in some notes or questions they have for their viewers.

We would like YouTube content creators to add their course content on our platform along with notes and quizzes for their viewers so that users can stick to a teacher of their choice and follow their entire course.

**## Ai used - Chat GPT**

    Prompts used: what can a user who is a beginner in coding expect from an e learner web application?

**## Team Members**:

Sukhwinder Singh 21BCS8221

Sakashi Thakur 21CBS1072

Shameen Ahmad 21BCS3807

Harsh Pandey 21BCS4517

Chander Thakur 21BCG1101

Rahul Ranjan 22BAI71369