***SOFTWARE REQUIREMENT SPECIFICATION***

***INTERACTIVE PROGRAMMING LANGUAGE LEARNER***

Team Members

M.Malarvannan ( 2022503011 )

Kalaidharun ( 2022503009 )

Table of Contents

**1.Introduction3**

1.1 Purpose of this Document3

1.2 Overview3

**2. Product Description3**

**3. Intended Audience 3**

**4.Requirements3**

4.1 Functional Requirements4

4.2 Non Functional Requirements5

**5.Assumptions and Options5**

5.1 Front End5

5.2 Back End5

5.3 Database6

**6.Use Case Diagram6**

1. **Introduction:**

**1.1 Purpose of this document:**

This SRS document comprises product description, intended audience, requirements both functional and non functional, and assumptions and options available for our Interactive Programming Language Learner.

**1.2 Overview:**

In today’s digital age, coding has become an indispensable skill, yet traditional learning methods often make the journey, challenging especially for beginners. The complexities of syntax, the dryness of theoretical lessons, and the lack of immediate feedback can be discouraging. Recognizing these challenges, we introduce the Interactive Programming Language Learner (IPLL), a revolutionary platform designed to reshape the coding education landscape.

1. **Product Description:**

In the rapidly evolving tech landscape, learning programming languages effectively and engagingly remains a significant challenge.

The “Interactive programming Language Leaner” project aims to bridge this gap by creating a dynamic platform that revolutionizes how individuals learn programming languages.

By integrating interactive coding challenges and feedback, this platform will transform the learning journey into an engaging and productive experience, equipping learners with the skills they need to excel in the programming world.

IPLL is intended to replace the dry learning environment of Programming languages with an interactive with an interactive, playful, and colorful environment.

1. **Intended Audience:**

Our product will be very useful for beginners, early coders, active learners, and passionate coders.

1. **Requirements:**

**4.1 Functional Requirements:**

1. Providing Choosing the programming language.
   1. We will be providing the user an option to choose the programming language he wish to learn.
   2. By this option the user can choose his desired language and start to embark his coding journey.
2. To allow user to engage in activities like learn and attend test
   1. The user can either learn a language or he can test his knowledge he posses in the programming language.
   2. In learn we will be teaching the user basic syntax and the path of the programming language
   3. In test section the user will be able to attend test based on his progress level. With this progress level the user will be prompted with the difficulty of the problem
3. Provide a interactive session.
   1. The learning process will be very much interactive as we would try to gamify the learning process, making it colorful, interactive, visually appealing.
4. Teach user from basic to advance concept.
   1. We would teach the user from the very basic syntax and coding modules, as he progress through our learning process, we would teach him advance components.
5. To conduct small quiz
   1. There’s no use in just learning. One would constantly need to test his might in order to excel in his field.
   2. With the above mentioned point on point, we will be conducting small quiz at the end of every lesson.
6. To Analyse Progress based on previous test result.
   1. We will be implementing an analyse mechanism where the user will be analysed based on his progress which uses previous test results.
7. Provide problem based on progress.
   1. As said earlier the user will be provided with problems based on his progress in the test section.
8. Provide daily question
   1. We will be using any API or we will manually integrate an option to display the user daily questions. With this the user will be able to increase his knowledge as he practice regularly.
9. To check overall progress
   1. The user will be able to check his progress, both learning and test.
10. To provide Badges
    1. Based on the progress the user will be given badges which would be helpful for him to stay motivated..

**4.2 Non functionality Requirements:**

1. Usability(user friendly interface)
   1. The interface should be intuitive, making navigation easy for users of all skill levels.
   2. Consistent layout and design patterns will improve the user experience.
2. Performance(Fast loading times)
   1. The platform should respond quickly.
   2. Efficient use of resources, minimizing unnecessary server requests and optimizing code.
3. Reliability
   1. The system should have minimal downtime, aiming for high availability.
   2. Fault tolerance mechanism
4. Portability.
   1. The code should be easily adaptable to different environments.

**5.Assumptions and Options:**

* It is assumed that this product will be very useful to freshers and soon will reach many.
* **5.1 Front end**
  + We have initially planned to use **React** for the front end
  + Alternatively, we have kept **html, CSS, JS** as a backup option
  + If possible, we will be implementing the entire project using **JAVA.**
* **5.2 Back end**
  + We have planned to use JAVA Spring Boot for backend service
  + Alternative: Java with **JavaFX**
* **5.3 Database**
  + If there is a need for database connection, we will be choosing **SQL** as our first option
  + Alternative: Mongo DB

**6.Use Case Diagram:**

