**Software Requirements Specification**

**for**

Intuicode

**Version 1.00 approved**

**Prepared by Team Brilliance**

**10/6/24**

**Table of Contents 2**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Jakob Sholler | 10/9 | Reasons for changes, Compiling work together. In appropriate Sections | 1.01 |
| Jakob Sholler | 10/15 | Recreated file to more resemble the sample as well as to Properly finish Missing sections for deliverable | 1.02 |
| Jakob Sholler | 10/19 | Moved over Ryans mock ups, then moved over Requirements numbers in the correct section, then filled in the requirements based on feedback in Project Feedback and Grading Document.  Moved Rest of Requirements over that were missing and renumbered, see key at bottom for Map for original Dr greg copy, and Current.  Added appendix D for Miscelaneous information, added key for NFREQ |  |
| Ryan O’Sullivan | 11/4 | Updated Section 4, providing more detail on the function requirement | 1.03 |

# **Introduction**

## **Purpose**

*This Document is for release of IntuiCode 1.00. An App to help beginner programmers learn to read/understand code. It also contains some tools/features to help strengthen the skills being learned, such as Little Tests/ Problems they can either Work on to solve, or View the solution to See how that code works. This Document also contains all various requirements for intuiCode, Including the scope audience, Requirements and Non Functional Requirements, Description of Intuicode Mockups for interfaces.*

## **Document Conventions**

*In comments On sidebar* ***Origin: Means*** *the original Author of the comment, Brought over from original document before reformatting. See Dr.Greg copy in Drive for original Comments and document.*

## **Intended Audience and Reading Suggestions**

*The intended audience is for the Developers, Project manager, Designers, and Dr.Greg*

## **Product Scope**

*IntuiCode is designed to support beginner programmers in learning how to read and interpret code, while also improving their programming skills. By offering interactive tools and resources, the application provides users with Step by step guidance, code examples, and feedback on their programs. Some key features include an integrated code editor, Contextual explanations, and interactive quizzes. The app breaks down coding concepts into more manageable tasks, allowing the users to progress at their own pace. Additionally, the interactive problems and quizzes reinforce learning by providing instant feedback for the user.*

## **References**

*Team Brilliance’s GIthub :* [*GitHub Link*](https://github.com/COS420-Fall24/TeamB)

*Dr Greg copy (Old SRS Before reformat/finishing missing sections)* [*Old SRS link*](https://docs.google.com/document/u/1/d/1n7FieBVICYrf373httsg8hr3WLywheKZkUMTriiVvuw/edit)

# **Overall Description**

## **Product Perspective**

*IntuiCode is an Independent self contained product its Purpose is to assist Beginner learners of programming to learn, A Few of the features are Community based forums to ask for help or to aide, Colored syntax for easier Identification of code segments, Built in Resources.*

## **Product Functions**

*\*Create An Account / Log in to Save or View current or previous progress from the app.*

*\* Integrated code Editor: Having an integrated code editor, saves time, as well as allows for tests/quizzes to be given in the app.*

*\*Resource Book*

*\*Fill in others when task is done.)\_S\_D0ASD\_SADAS)\_DS\_FSJAf-aFA(weird to make it easy to see)*

## **User Classes and Characteristics**

*One Class is logged in users This should be another large class that will Have all functionality for the app, including anything that requires data to be saved.*

*Another is Non-logged in users. This will be A large Class, This will be the most used class, although hopefully logged in will be. Users who are not logged in will have access to Almost all functionality. They should have access to any core functionality that does not require saving data.*

*The Dev/Admin Class will have all the features/Characteristics of the other classes, e.g Accessing/ Saving Coursework. They will also need a higher level of security, Since they will have the ability to modify the contents of the app, and access any data or system logs.*

## **Operating Environment**

*IntuiCode is Running on React, which includes HTML, CSS and Javascript. It should Run on Any up to date Major Browsers Chrome, Firefox, Microsoft Edge with React Installed and JavaScript Enabled.*

## **Design and Implementation Constraints**

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).> FILL IN LAter*

## **User Documentation**

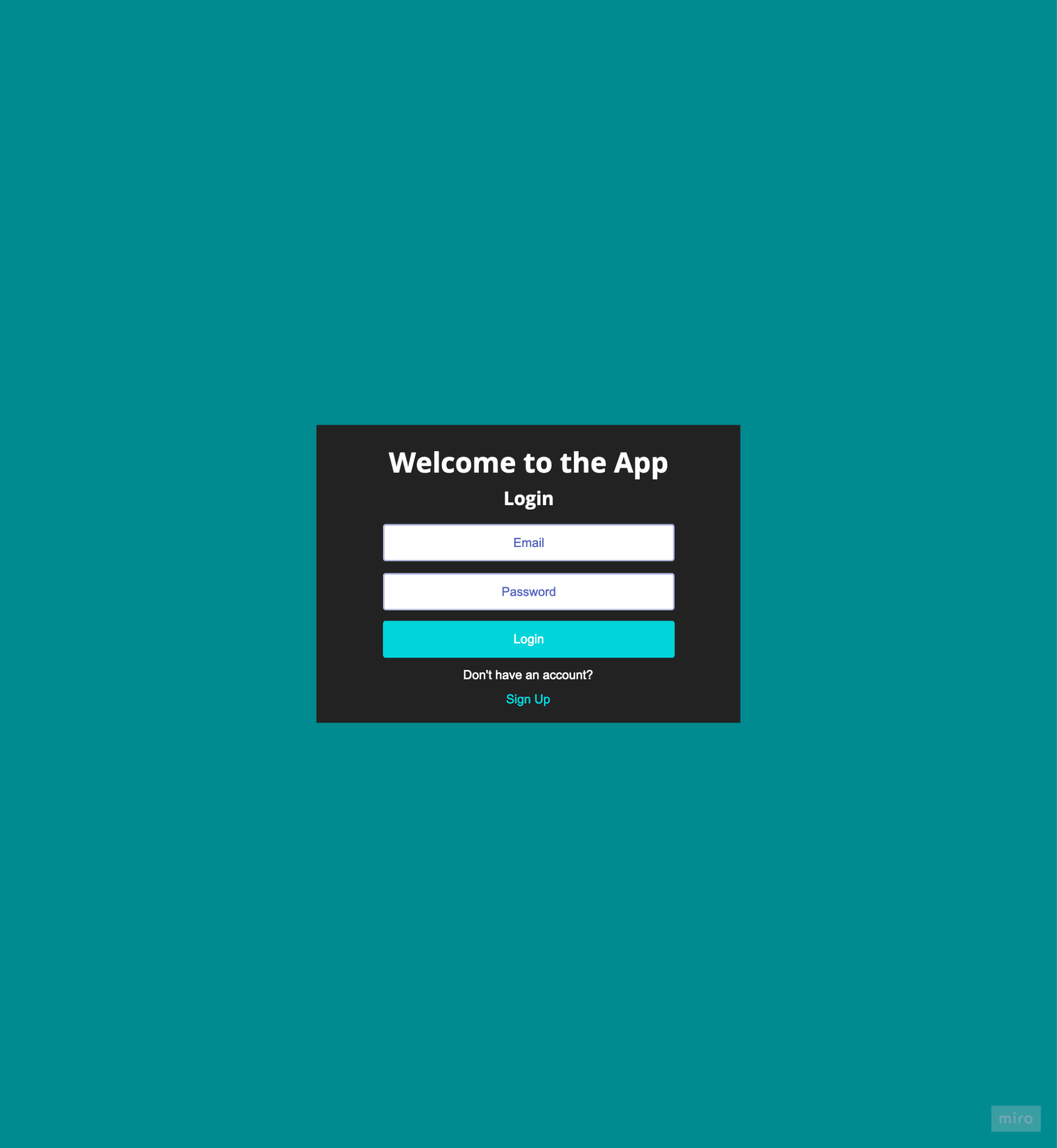
*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>FILL IN LATER*

## **Assumptions and Dependencies**

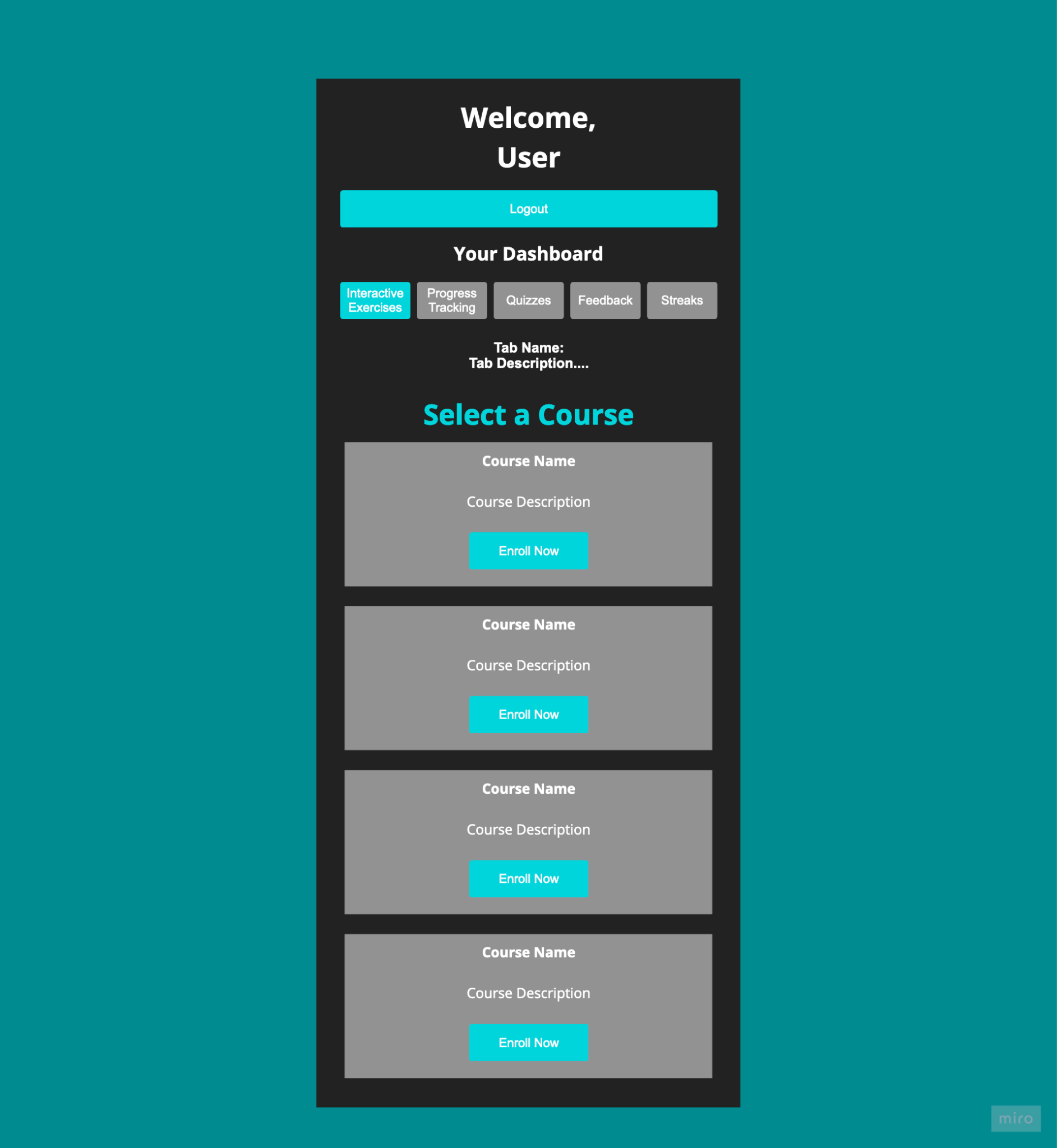
*Intuicode is developed using react, Which includes HTML, CSS and javascript, If javascript is disabled the app will not run*

# **External Interface Requirements**

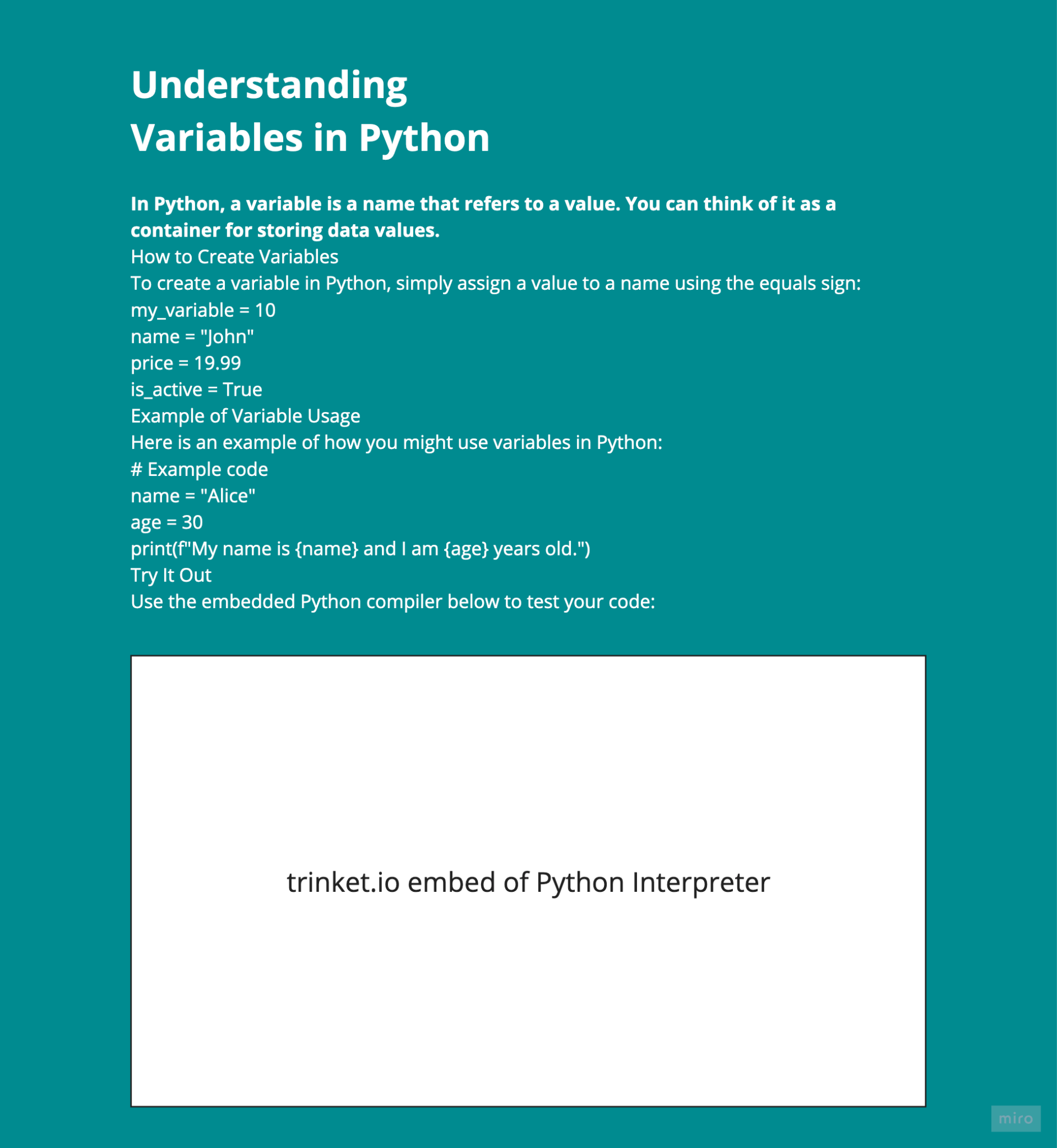
## **User Interfaces**



The Landing page when you are not logged in, prompting the user to log in.

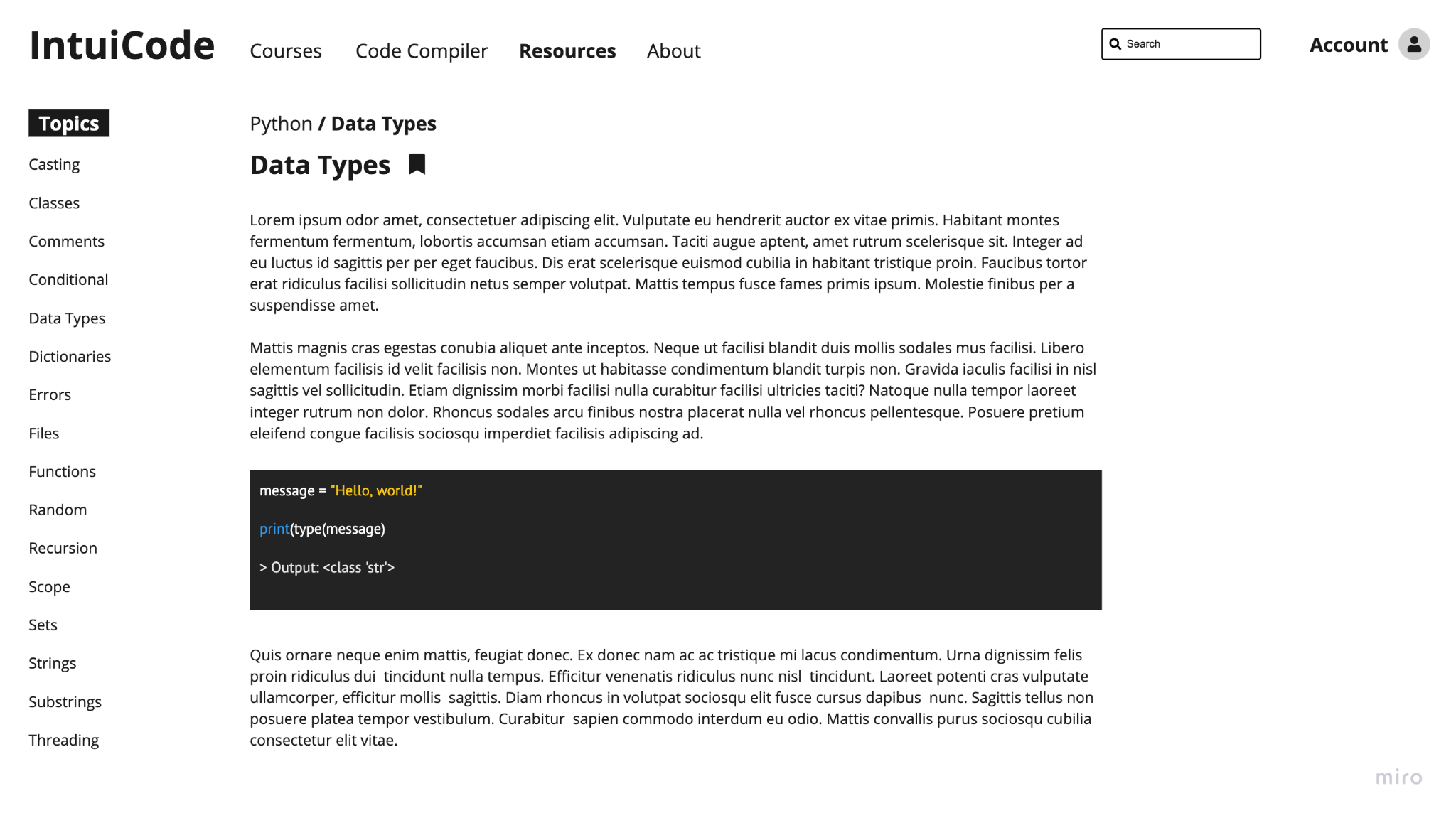


The landing page when you’re logged in.  
Top to Bottom: Buttons that allow you to switch tabs of what to do. A description of the current tab. The courses for the current tab

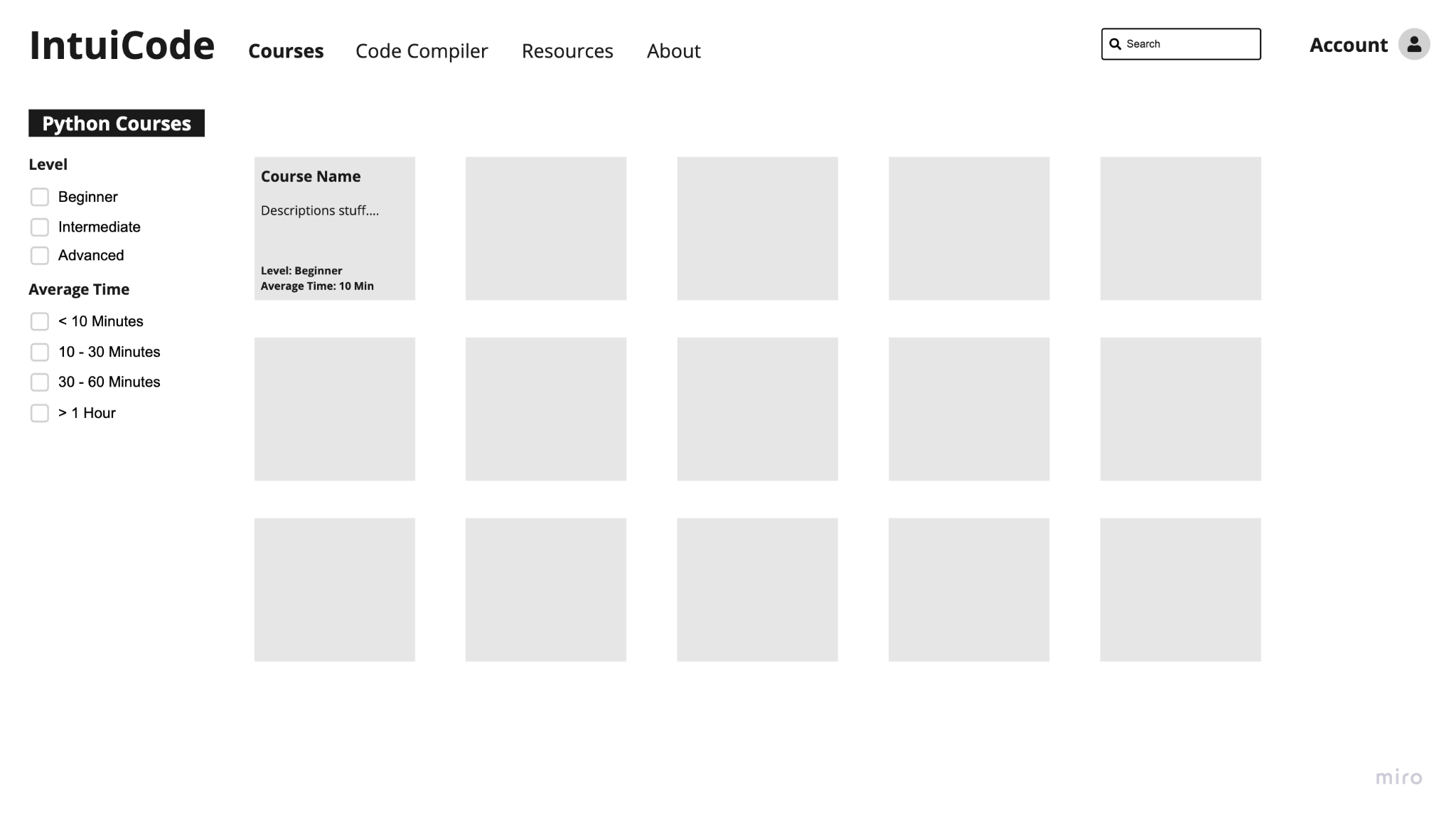


The Code Compiler for Challenges and Projects

Top to Bottom: Name and description of the Selected Course, and the Embedded trinket.io Python Interpreter to code the above problem.



The Resources tab, with a list of different topics to read about.



The Courses tab that allows you to select a specific topic to learn. It also allows you to sort by skill level and course time.

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>FILL IN*

## **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.> FILL IN*

# **System Features**

## Quizzes and Feedback

**4.1.1 Description and Priority**

*The system provides quizzes and tests on programming topics, with immediate feedback on user answers (correct/incorrect). Priority: High.*

**4.1.2 Stimulus/Response Sequences**

* ***Stimulus****: User submits an answer to a quiz question.*
* ***Response:*** *System indicates whether the answer is correct or incorrect.*
* ***Stimulus:*** *User finishes a quiz.*
* ***Response:*** *System displays the user's score and detailed feedback for incorrect answers.*

**4.1.3 Functional Requirements**

*REQ-1: The system must provide quizzes and tests on programming topics.*

*REQ-2: The system must give feedback (Correct/Incorrect) after each question.*

## Interactive Learning Features

**4.2.1 Description and Priority**

*The app provides challenges, and progress tracking, including personalized paths. Priority: High.*

**4.2.2 Stimulus/Response Sequences**

* ***Stimulus****: User completes a challenge*
* ***Response:*** *System provides feedback and achievements to the user.*
* ***Stimulus:*** *User goes to the dashboard.*
* ***Response:*** *System displays completed lessons, streaks, and progress tracking.*

* ***Stimulus:*** *User selects a learning path.*
* ***Response:*** *System recommends lessons and challenges tailored to the user.*

**4.2.3 Functional Requirements**

*REQ-9: The system must provide a point or reward system for daily activity.*

*REQ-10: The system should track consecutive days of user engagement.*

*REQ-11: The system must allow users to review past sections.*

*REQ-12: The system should provide summaries from previous lessons.*

*REQ-13: The system must provide encouraging messages after the user completes tasks.*

*REQ-14: The system should reward users with achievements.*

*REQ-15: The app shall have lessons including explanations, examples, and interactive coding exercises.*

*REQ-16: The app shall allow users to write and test code directly within the lesson.*

*REQ-17: The app shall offer coding challenges at varying difficulty levels based on the user’s level.*

*REQ-18: The app shall provide daily challenges to warm up the user prior to working on the lessons.*

## Code Editor Features

**4.3.1 Description and Priority**

*The system provides a built-in coding environment with tools for testing, debugging, and personalization. Priority: Medium.*

**4.3.2 Stimulus/Response Sequences**

* ***Stimulus****: User writes code in the editor.*
* ***Response:*** *System highlights syntax and displays warnings or errors.*

**4.3.3 Functional Requirements**

*REQ-33: The app shall provide a built-in coding environment where users can write, run, and debug their code.*

*REQ-23: The app shall provide coloring for functions, variables, and imports, making code easy to read.*

*REQ-24: The app should display error messages or warnings in the code editor.*

## Resources

**4.4.1 Description and Priority**

*The app provides a community forum, external resources, and bookmarking. Priority: Medium.*

**4.4.2 Stimulus/Response Sequences**

* ***Stimulus****: User posts a question in the forum.*
* ***Response:*** *System adds the post and notifies relevant users for help.*
* ***Stimulus****: User bookmarks a resource.*
* ***Response:*** *System saves the resource for later access.*

**4.4.3 Functional Requirements**

*REQ-21: The app shall provide a community forum where users can ask for help or help others.*

*REQ-22: The system should allow users to like forum posts made by others.*

*REQ-27: The app shall provide links to resources that can help users learn more.*

*REQ-28: The app shall allow users to bookmark resources for later reference.*

* 1. **Functional Requirements**

REQ-1 The System must provide quizzes and tests on programming topics.

REQ-2 The System must give feedback(Correct/Incorrect) After each question

REQ-3 The system must apply syntax highlighting to all supported programming languages.

REQ-4 The system must allow users to click on or search for functions and jump to their Definitions.

REQ -5 The system must provide keyboard shortcuts for navigating code sections.

REQ-6 The System should display notes in a way that does not interfere with the code view

REQ-7 The system should limit the number of pop-ups or notifications during use.

REQ-8 The System should limit the number of popups or notifications during use.

REQ -9 The system must provide a point or reward system on daily activity

Req -10 The system should track consecutive days of user engagement

REQ-11 The system must allow users to review past sections

REQ-12 The system should provide summaries from previous lessons

REQ-13 The system must provide encouraging messages after the user completes tasks

REQ-14 The system should rewards users with achievements

REQ-15 The app Shall have lessons including explanations, examples and interactive coding

exercises,

REQ-16 The app shall allow users to write and test code directly within the lesson.

REQ-17 The app shall offer coding challenges at varying difficulty levels based on the users

level

REQ-18 The app shall provide daily challenges to warm up the user prior to working on the

lessons

REQ9 The app shall include tutorials that users can follow to create small

projects(Calculator, little games, etc.

REQ-21 The app shall provide a community forum where users can ask for help or help

others

REQ-22 The System should allow users to like forum posts made by others.

REQ-23 The app shall provide coloring for functions, variables, and imports making it easy

to read

REQ-24 The app should also display error messages or warnings in the code editor.

REQ-25 The app shall track the users progress through lessons, quizzes, and challenges.

REQ-26 The app shall display a dashboard showing complete lessons Along with a streak

counter.

REQ-27 The app shall provide links to different resources that can help a user learn more.

REQ-28 The app shall allow users to bookmark resources for later reference.

REQ-29 The app shall offer users the option to choose a learning path based on their

interests and goals

REQ-30 The app shall recommend lessons and challenges based on their previous lessons.

REQ-31 The app shall include a feedback form for users to write any concerns they have

REQ-32 The app shall allow users to rate lessons and suggest improvements

REQ-33 The app shall provide a built-in coding environment where users can write, run, and

debug their code.

REQ -34 The app shall allow users to choose a language for the coding environment

*REQ -35 The app shall provide an internal search functionality that returns results within 2*

*seconds for content available within the app, such as language specific sections like arrays.*

# **Other Nonfunctional Requirements**

## **Performance Requirements**

*NFREQ-1 The system should be able to handle 1000 concurrent users with a response time of less than 2 seconds, 90% of the time.*

*NFREQ-2 The apple should function with all major browsers. Firefox, Chrome, Safari*

*NFREQ-3 The app shall maintain an uptime of 99.9% excluding scheduled maintenance periods notified in advance.*

## **Security Requirements**

NFREQ-4 The System Shall require a password minimum of 8 characters, and include at least 1 symbol, number or capital letter.

*NFREQ-5 The system Shall log users out after 2 hours of inactivity.*

*NFREQ-6 The app shall include Encryption for data transmission (Https)*

*NFREQ-7 The app Shall comply with GDPR regulations*

*NFREQ-8 The app shall include Secure User authentication.*

## **Software Quality Attributes**

*Usability*

*Performance*

*The app shall respond fast, With processing times of user requests of 2 seconds for 90 % of the time under peak loads.*

*Scalability*

*The system shall be designed to accommodate growth. Allowing of new features without significant impact on performance*

*Maintainability*

*The Code shall be designed to help with maintenance and future updates, saving time for developers.*

*Reliability*

*The system will maintain a high level of reliability, being 99.9% uptime outside of scheduled maintenance.*

# **Other Requirements**

*NFREQ-9 The System Shall provide a mechanism for users to report bugs and provide feedback*

*NFREQ-10 The system shall provide a guide on how to navigate the program after creating an account*

*NFREQ-11 The System shall send notifications to remind users to practice.*

*NFREQ-12 After Overview Guide/Training users can do tasks at at 90% correctness level. `*

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

*Appendix A: Glossary*

*Appendix B: Analysis Models*

*Index page numbers*

*3.2 Software Interfaces*

*3.3 Communication interfaces.*

*Appendix D: Other Miscellaneous*

| *SRS\_Template NFREQ #* | *Dr. Greg copy (Old Document NFREQ#* |
| --- | --- |
| *#1* | *#1* |
| *#2* | *#3* |
| *#3* | *#13* |
| *#4* | *#2* |
| *#5* | *#8* |
| *#6* | *#14* |
| *#7* | *#14* |
| *#8* | *#14* |
| *#9* | *#4* |
| *#10* | *#6* |
| *#11* | *#7* |
| *#12* | *# N/.A* |