Loom: One Web Development Tool for Everyone

***Abstract*—Loom is a desktop application designed to bridge the gap between beginner and expert web development tools. Existing tools in the domain provide either easy-to-use interfaces intended for novice users, sacrificing expressiveness, or advanced interfaces intended for developers who already know web technologies, but inaccessible by ordinary users. Loom is a first-of-its-kind tool which provides novice users the full power of web technologies such as HTML and CSS without the need to write code. In addition, Loom aims to impart the basics of these technologies to users as they work with the tool, teaching valuable concepts that can be carried over to other tools.**

I. Introduction

There are many existing web-development tools intended for a wide range of users. Some of these tools are targeted at novice users, and provide very easy-to-use interfaces with high-level features such as drag-and-drop and WYSIWYG editing. Other tools are intended for developers, and provide textual interfaces for working with code. Others still are intended for designers. The separation of these tools, and the incompatibilities between them, leads to several problems:

- Coordination between these groups is difficult. If a novice user using a tool such as Wix or Squarespace decides they need a developer to continue their work, existing work will likely be wasted as the developer chooses to use a different tool.
- The work done by designers in tools such as Figma, Sketch or Invision has to be mirrored by developers, wasting effort.
- Novice users are learning how to use a specific tool – not the underlying technologies. Experience with this tool likely will not carry over to another.

Some tools try to target more than one type of user. For example, Wordpress has a code editor, enabling developers to directly edit HTML code. Blocs, a desktop application, provides designer-friendly tools, WYSIWYG editing and templates for beginners, and code editing for developers. But one problem with these tools still remains: novice users may never learn how to use ”advanced” features such as code editing without doing their own research into web development.

II. Approach

Loom takes a different approach from most existing tools. It provides one common interface which is closely related to the underlying web technologies. This way, every user has the full power of these technologies. The remaining challenge is making the interface intuitive to beginners, and familiar to developers and designers. To accomplish this, Loom uses two strategies:

- **Provide helpful, simple abstractions over existing technologies:** working directly with web technologies can be frustrating, even for developers. Loom aims to provide a handful of simple, but useful, abstractions over these technologies to make working with them easier.
- **Provide a user interface that ”scales” with the user:** The interface should be easy for novices to pick up, but provide the full power for those who want it. For example, WYSIWYG editing and graphical components such as color pickers should make styling easy for beginners, but developers should be able to enter textual values for these properties as well.

A. Goals

Loom has the following motivating goals:

- TODO

(what are the motivating goals behind the design of Loom; e.g. easy to use, teaches the user basic web technologies, easy for developers to pick up)

III. Design

(UI design, inputs / outputs, etc)

IV. Implementation

(more inner details of how Loom works; e.g. high-level code structure, usage of event listeners)

V. Evaluation

(how well does Loom achieve its goals? e.g. user testing results)

VI. Conclusion

(Did we achieve what we set out to do? What issues remain? What future work is there?)

Acknowledgment

The authors would like to thank...