

Glimpse

Feasibility Report

Group – 2

1. Raman Sharma – 22114076

Contact No: +91 75218 29455

Email ID: raman_s@cs.iitr.ac.in

2. Boda Yashwanth – 22114022

Contact No: +91 89779 25111

Email ID: boda_y@cs.iitr.ac.in

3. Ayush Ranjan – 22114018

Contact No: +91 74829 58551

Email ID: ayush_r@cs.iitr.ac.in

4. Souvik Karmakar – 22114096

Contact No: +91 86429 24659

Email ID: souvik_k@cs.iitr.ac.in

5. Sarvasva Gupta – 22114086

Contact No: +91 79899 04811

Email ID: sarvasva_g@cs.iitr.ac.in

6. Anvit Gupta – 22114009

Contact No: +91 94620 11044

Email ID: anvit_g@cs.iitr.ac.in

7. Vineet Kumar – 22114107

Contact No: +91 92634 36403

Email ID: vineet_k@cs.iitr.ac.in

Project Overview

The project Glimpse is a chrome extension which aims to enhance user productivity, navigation, and search experience. It provides an entirely new user experience by displaying motivational quotes, dynamic to-do list, organized bookmarks and customisable wallpaper on the home page. Also, it provides other useful features like scrollable website preview upon hovering on the link, displaying the estimated reading time, ability to bookmark the website and open the website in full tab as and when the user wishes. There are also additional features like grouping of bookmarks using sentiment analysis for efficient organization of bookmarked links, summary of the content in website and customisable preview size.

Customer

The project Glimpse is useful for the following customers in real-life:

- **Students and Researchers:** This group often juggles multiple tasks, conducts research, and manages large bookmark lists, making them ideal candidates for the organization and navigation features.
- **Professionals:** Individuals working in fast-paced environments can benefit from improved time management tools like to-do lists and simplified web previews to stay focused and efficient.
- **Casual Internet Users:** Anyone who uses Chrome for browsing and needs help with organization, managing bookmarks, and quickly evaluating websites can benefit from this extension.

Dummy Customer: Sarah, the Information Architect

Demographic Information:

- Age: 25
- Occupation: Information Architect
- Location: Suburban

Psychographic Information:

- Values efficiency and clear information hierarchy.
- Seeks tools that enhance information organization and streamline content consumption.

Needs and Pain Points:

- Often encounters information overload while researching online.
- Struggles with managing and categorising a vast number of bookmarked websites.

- Finds it time-consuming to evaluate the relevance of search results before opening them.

Behaviours and Preferences:

- Actively seeks out tools that improve information organisation and content consumption.
- Prefers intuitive interfaces with clean aesthetics and customizable features for personalised workflows.

User Goals and Expectations:

- Aims to improve research efficiency and reduce wasted time evaluating irrelevant websites.
- Expect Glimpse to provide clear website summaries, facilitate bookmark organisation, and offer a clutter-free browsing experience.

User Experience Considerations:

- Values clear presentation of information with a focus on usability.
- Seeks a visually appealing extension that integrates seamlessly with her existing workflow.

This revised version highlights the target customer segments and provides a more detailed dummy customer persona named Sarah, reflecting the needs and preferences of information architects like her.

Visibility Plan

Effective communication with the customer and Supervisor is crucial for the success of this project. Regular progress reports will be shared via weekly meetings, emails, and a shared project management platform, such as Trello or Asana. The team will also maintain an open line of communication through a dedicated project email address and instant messaging platforms for quick updates and issue resolution.

Statement of the Task

The task of this project is to develop a Chrome extension named Glimpse that enhances user productivity, navigation, and search experience by offering following features:

- **New Tab features:** Personalized quotes, dynamic to-do lists, curated bookmarks, and customizable wallpapers.

- **Website Preview features:** Scrollable content preview, bookmarking, estimated reading time, optional simplified view, and (optionally) AI-powered content summaries, sentiment analysis, image/video previews, and resizable preview window.

It involves developing HTML, CSS and JavaScript for the extension's interface, utilising Chrome APIs wherever required, integrating the AI/ML service for content summarization, develop server-side components, etc.

Preliminary Requirements Analysis

Functional Requirements:

- **New Tab Functionality:**
 - Display personalized quotes with user names.
 - Manage and display a dynamic to-do list with functionalities like adding, editing, and marking tasks complete.
 - Implement a curated bookmark list with the ability to add, edit, and delete bookmarks.
 - Provide options to upload or choose from a library of wallpapers and enable daily wallpaper changes.
- **Website Preview Functionality:**
 - Implement a scrollable preview window displaying the main content of websites.
 - Offer a button to open the entire webpage in a new tab.
 - Integrate a bookmarking feature for websites from the preview.
 - Calculate and display an estimated reading time based on content length and average reading speed.
 - (Optional) Develop a simplified view that removes unnecessary elements from the preview.
- **Advanced Features (Optional):**
 - Integrate AI/ML services for content summarization and sentiment analysis of webpages.
 - Display prominent images and videos from the webpage within the preview.
 - Allow users to customize the size of the preview window.

Non-Functional Requirements:

- **Usability:** The extension should be user-friendly and intuitive, with clear instructions and easy navigation.
- **Performance:** The extension should load quickly and operate efficiently without significantly impacting browsing speed.

- **Compatibility:** The extension should be compatible with various Chrome versions and operating systems.
- **Security:** The extension should handle user data securely and adhere to Chrome extension security guidelines.
- **Maintainability:** The code should be well-documented, modular, and easy to maintain for future updates and bug fixes.

Suggested Deliverables

- **Developed Chrome extension:** This is the core product, encompassing all functionalities outlined in the project plan, including:
 - **New Tab features:** Personalized quotes, dynamic to-do list, curated bookmarks, and customizable wallpapers.
 - **Website Preview features:** Scrollable content preview, bookmarking, estimated reading time, optional simplified view, and (optionally) AI-powered content summaries, sentiment analysis, image/video previews, and resizable preview window.
- **Documentation:**
 - User manual explaining how to install and use the extension.
 - Developer documentation outlining the code structure, functionalities, and API integrations.
- **Testing reports:** Documentation summarizing the testing process, identified bugs, and implemented fixes.

Project Methodology

We will follow an iterative development approach, combining elements of the modified waterfall model and iterative refinement. The project will be divided into several phases: requirements analysis, design, development, testing, and deployment. Each phase will undergo multiple iterations, allowing continuous improvement based on user feedback and evolving requirements.

Outline Plan and Milestones

Phase 1: Planning and Requirements Gathering

- **Milestone 1.1:** Finalize core functionalities and features based on user needs and technical feasibility.
- **Milestone 1.2:** Define technical specifications, development tools, and libraries.
- **Milestone 1.3:** Create user interface mock-ups and prototypes for visual representation.

Phase 2: Development

- **Milestone 2.1:** Develop core functionalities of the new tab features (quotes, to-do list, bookmarks, wallpapers).
- **Milestone 2.2:** Implement website preview functionality (scrollable content, open in new tab, bookmarking).
- **Milestone 2.3:** Integrate additional features (estimated reading time, simplified view) if applicable.
- **Milestone 2.4:** Develop backend components for AI/ML integration or other server-side functionalities (optional).

Phase 3: Testing and Refinement

- **Milestone 3.1:** Conduct thorough unit testing to ensure individual components function as intended.
- **Milestone 3.2:** Perform integration testing to verify interaction between different functionalities.
- **Milestone 3.3:** Conduct user testing to gather feedback on usability, performance, and overall user experience.
- **Milestone 3.4:** Refine the extension based on user feedback and identified bugs.

Phase 4: Documentation and Deployment

- **Milestone 4.1:** Create user documentation
- **Milestone 4.2:** Prepare installation packages

Phase 5: Final Testing and Delivery

- **Milestone 5.1:** Perform final testing of the product
- **Milestone 5.2:** Deliver the completed application to the customer

Risk Analysis

Technical Risks:

- **Feature implementation issues:** Unexpected technical challenges may arise during development, causing delays and requiring innovative solutions.
- **Integration challenges:** Integrating external services (e.g., AI/ML) can be complex and might not function seamlessly, impacting functionalities and user experience.
- **Security vulnerabilities:** Unforeseen security vulnerabilities in the extension could expose user data or compromise their browsing experience.

Operational Risks:

- **Project delays:** Unforeseen challenges during development, testing, or review stages can lead to delays in launch and impact project timelines and budget.
- **Team resource constraints:** Limited developer bandwidth or expertise could hinder progress and require additional resource allocation or adjustments to the project scope.
- **Third-party service outages:** If the extension relies on external services (e.g., content summarization), outages or service disruptions could affect functionality and user experience.

Probable Technical Requirements

Frontend:

- **Development languages:** HTML, CSS, and JavaScript
- **Chrome extension APIs:**
 - OpenWeather API: To access current weather and forecast, to display in the home page.
 - chrome.storage API: To allow the chrome extension to store, retrieve and track changes to user data.

Backend:

- **Server-side language:** Depending on the chosen approach, languages like Python and Node.js could be used for server-side functionalities.
- **Database:** If needed, a database like MongoDB or PostgreSQL could store user preferences or data processed on the server-side (e.g., content summaries).
- **AI/ML integration (optional):** If using AI/ML features like content summarization or sentiment analysis, integration with an AI/ML service through their APIs would be required.

Additional Considerations:

- **Version control system:** Use a version control system like Git for code management, collaboration, and tracking changes.