PES UNIVERSITY, Bengaluru
Department of Computer Science and Engineering
B. Tech (CSE) – 5th Semester – Aug-Dec 2023

B.TECH. (CSE)
V Semester
UE21CS341A –Software Engineering

PROJECT REPORT
on

Chrome Extension
All-in-One Productivity Enhancer

Submitted by :

PES2UG21CS482	Sarthak S Kumar	PES2UG21CS474	Sanath Kumar R
PES2UG21CS484	Sathish Kumar G	PES2UG22CS822	Sumukha N M

August – Dec 2023
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BENGALURU – 560100, KARNATAKA, INDIA

Table of Contents

Project Proposal

Proposed Project Description:

Our project aims to create an all-in-one productivity-enhancing Chrome extension that combines various functionalities to streamline and optimize users' browsing experiences. This multifaceted extension will empower users to manage their web activities more efficiently, incorporating features like a Bookmark Manager, Language Translator, Quick Notes and To-Do Lists, Adaptive Dark Mode, Web page Annotations, Password Generator and Manager, Quick Search AI, and a Quick Screenshot tool.

Functional Features and Qualitative Properties:

1. Bookmark Manager:

- Allow users to save, categorize, and manage their favorite websites and resources.
- Functional Features:
- Bookmark creation and organization.
- Folders and tagging system for categorization.
- Import/export bookmarks.

2. Language Translator:

- Integrate a language translation tool that can translate selected text or entire web pages into different languages.
- Functional Features:
- Text and webpage translation.
- Multiple language support.

3. Quick Notes and To-Do Lists:

- Provide users with a simple note-taking and to-do list feature for quick and efficient task management.
- Functional Features:
- Create, edit, and delete notes and tasks.
- To-do list organization.

4. Adaptive Dark Mode:

- Automatically enable dark mode for websites that lack native support, enhancing nighttime browsing.
- Functional Features:
- Dark mode activation for supported websites.
- User-configurable settings.

5. Webpage Annotations:
- Enable users to annotate and highlight web content, facilitating research and study online.
- Functional Features:
- Highlighting, underlining, and note-taking on web pages.
- Export and share annotations.

6. Password Generator and Manager:
- Offer tools to generate strong passwords and manage them securely.
- Functional Features:
- Password generation with customizable options. - Encrypted password storage.

7. Quick Search AI:
- Implement a quick web search functionality powered by AI, allowing users to find information instantly.
- Functional Features:
- Instant web search within the extension.
- Relevant search results based on AI analysis.

8. Quick Screenshot:
- Allow users to capture quick screenshots of web pages and save or share them instantly.
- Functional Features:
- Screenshot capture and saving.
- Annotation on screenshots.
- Screenshot organization and history.

Plan of Work and Product Ownership:
1. Sarthak S Kumar (SRN: PES2UG21CS482)
- Functional Feature: Bookmark Manager, Quick Search AI

2. Sathish Kumar G (SRN: PES2UG21CS484)
- Functional Feature: Language Translator, Password Generator and Manager

3. Sanath Kumar R (SRN: PES2UG21CS474)
- Functional Feature: Quick Notes and To-Do Lists, Quick Screenshot

4. Sumukha N M (SRN: PES2UG21CS822)
- Functional Features: Adaptive Dark Mode, Webpage Annotations.

Project Plan

SDLC Model

1. Implementing the project using Lean Agile:

Implementing the project using Lean Agile is a wise choice, as it emphasizes efficiency, adaptability, and delivering value to the end-users early and often. Let's align the Lean Agile approach with the project components.

Lean Agile Principles and Project Components:

- **Iterative and Incremental Delivery:**

Break down the project into small, manageable increments, delivering a potentially shippable product after each sprint (e.g., 2-4 weeks per increment). Each increment should have a set of features that add value to the end-users.

- **Feedback Loops and Continuous Improvement:**

Encourage frequent feedback from stakeholders, end-users, and team members. Use this feedback to adapt and improve the product continuously.

- **Empowering Teams and Individuals:**

Encourage teams to self-organize and make decisions. Trust their expertise and empower them to find the best ways to implement the functionalities assigned to them.

- **Value Stream Mapping:**

Map the value stream for each feature to identify bottlenecks, delays, or unnecessary steps in the development process and work on streamlining the workflow.

- **Visual Management and Information Radiators:**

Use visual tools like Kanban boards or Scrum boards to visualize the workflow, track progress, and make the project's status transparent to all stakeholders.

Aligning Lean Agile with Project Components:

- **Sprint Planning:**

Conduct sprint planning sessions at the beginning of each sprint to select and prioritize features from the backlog for that sprint.

- **Daily Stand-ups:**

Hold daily stand-up meetings to discuss progress, challenges, and plan for the day's work, ensuring everyone is aligned and aware of the project's status.

- **Sprint Reviews and Retrospectives:**

At the end of each sprint, conduct a review meeting with stakeholders to showcase completed features and gather feedback. Also, conduct a retrospective to reflect on the sprint and identify areas for improvement.

- **Minimum Viable Product (MVP):**

Identify and prioritize a Minimum Viable Product (MVP) that includes essential features to deliver value early to the users.

- **Kanban for Workflow Management:**

Use Kanban boards to visualize the workflow, manage tasks, and optimize the development process.

- **Continuous Integration and Deployment:**

Implement continuous integration and deployment practices to ensure that new features are integrated smoothly and deployed regularly.

By aligning the Lean Agile approach with the project components and using principles such as iterative development, feedback loops, and continuous improvement, you can efficiently deliver a high-quality product in a lean and agile manner, meeting the users' needs effectively.

2. Recommended Tools for Software Development Lifecycle:

Planning and Project Management:
Planning Tool: Jira for creating and managing tasks, user stories, and sprints.
Communication and Collaboration: Slack or Microsoft Teams for team communication and collaboration.

Design and Prototyping:
Design Tool: Figma for designing the user interface and creating prototypes.

Version Control and Collaboration:
Version Control: Git for version control and GitHub or GitLab for collaborative development and hosting repositories.

Development:
IDE (Integrated Development Environment): Visual Studio Code
Programming Languages: Depending on the project, appropriate programming languages like JavaScript, HTML, CSS, etc., will be used.

Bug Tracking and Issue Management:
Bug Tracking: GitHub for tracking bugs and issues.

Testing:

- Automated Testing: Selenium or Cypress for automated functional testing.
- Unit Testing: Jest for JavaScript unit testing, JUnit for Java, or NUnit for .NET.
- Performance Testing: Apache JMeter for performance testing.
- Continuous Integration and Deployment (CI/CD): Jenkins, Travis CI, or GitLab CI/CD for automated build, test, and deployment processes.

These tools will help in effectively managing the project, designing the user interface, ensuring version control and collaboration, facilitating development, tracking bugs, and conducting testing throughout the software development lifecycle. The specific tools chosen may vary based on the team's preferences, project requirements, and technology stack being used.

3. Software Development Project Deliverables:

Reusable Components:

3.1 Email-Password Authentication Module:

- Justification: This component can be reused in other projects that require email-password authentication, promoting security and standardization across applications.

3.2 Google Login Integration Module:

- Justification: Reusable as it provides a standardized way to integrate Google login, potentially used in multiple projects that require Google authentication.

3.3 ChatGPT Integration Module:

- Justification: Can be used in other projects that require integration with ChatGPT or similar chatbot APIs, facilitating natural language interactions in various applications.

3.4 File Compression API Integration:

- Justification: This can be reused in projects that need file compression functionality, saving development time and effort in future endeavors.

3.5 Notes Management Module (CRUD operations and Database integration):

- Justification: CRUD operations for notes and database integration can be a common requirement across many applications, making this component highly reusable.

3.6 Text-to-Speech Integration Module:

- Justification: Reusable in projects requiring text-to-speech functionality, providing a standardized approach to converting text to audio.

3.7 Color Picking Interface and Integration:

- Justification: The color picker interface and integration can be used in any project that requires color selection, saving development time.

3.8 Password Saving Interface and Integration:

- Justification: Reusable in projects needing password saving functionality, allowing for a standardized approach to password management.

3.9 Translation API Integration:

- Justification: Can be reused in various projects that require translation services, reducing the effort required to integrate translation functionality.

3.10 Calculator Interface and Integration:

- Justification: Can be reused in projects that need calculator functionality, providing a standardized calculator interface and integration.

Build Components:

3.1 Unique Sessions and Chat History for ChatGPT:

- Justification: This is specific to the ChatGPT integration in this project and not likely to be reused as it's tailored to the project's unique requirements.

3.2 Integration with Web System API for Screenshots:

- Justification: Custom integration for screenshots, specific to this project's requirements, not likely to be reused as it's tailored to this project.

3.3 Integration for Clearing Browsing History:

- Justification: Project-specific integration for clearing browsing history, not likely to be reused in other projects.

Notes:

The "reusable components" are generic functionalities that can be used in a variety of applications and are not tied to the specific use case of this project. They can enhance productivity and maintain consistency across different projects.

The "build components" are project-specific integrations or functionalities necessary for this particular project. They may not be applicable or beneficial in other projects without significant modification.

These categorizations will guide future project planning, helping to identify components that can be leveraged in subsequent projects for improved efficiency and consistency.

Estimation of Work

### 1. *User Authentication (0.5 person-month)*
- Email-password authentication (Integrating off-the-shelf API)
- Google login integration (Using Google API)

### 2. *Screenshot Functionality (0.5 person-month)*
- Integrating Web System API for screenshots

### 3. *ChatGPT Integration (1 person-month)*
- Setting up unique sessions and chat history
- Integrating ChatGPT API

### 4. *File Compressor (0.5 person-month)*
- Integrating file compression API

### 5. *Notes Taking Feature (1 person-month)*
- Designing the user interface for notes
- Integrating CRUD operations using off-the-shelf API
- Database integration for storing notes

### 6. *Text-to-Speech (0.5 person-month)*
- Integrating text-to-speech API
- Designing user settings for voice and speed

### 7. *Color Picking Feature (0.5 person-month)*
- Designing the color picker interface
- Integrating color addition to web pages using off-the-shelf API

### 8. *Clearing Browsing History (0.5 person-month)*
- Integrating browsing history clearing API

### 9. *Auto Save Password (0.5 person-month)*
- Designing the password saving interface
- Integrating auto-save password functionality using off-the-shelf API

### 10. *Translation Tool (0.5 person-month)*
- Integrating translation API

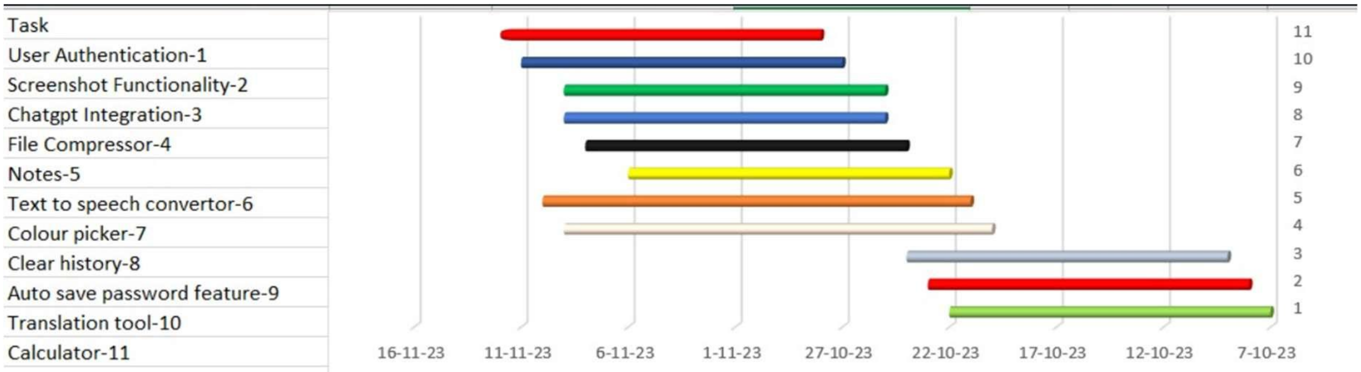
### 11. *Calculator (0.5 person-month)*
- Designing the calculator interface
- Integrating calculator functionality with history using off-the-shelf API

### *Total Estimated Effort: 7 person-months*

Considering a team of four developers, the project can be completed in approximately *1.75 months*.

Please note that the actual integration process can vary based on the complexity of the APIs and the specific requirements of your project. Also, ensure that you comply with the terms of service of the APIs you plan to use.

Gantt Chart Visualization



Design Diagrams

SDLC Model:
Implementing the project using Lean Agile is a wise choice, as it emphasizes efficiency, adaptability, and delivering value to the end-users early and often.

Lean Agile Principles and Project Components:

- **Iterative and Incremental Delivery:**

Break down the project into small, manageable increments, delivering a potentially shippable product after each sprint (e.g., 2-4 weeks per increment). Each increment should have a set of features that add value to the end-users.

- **Feedback Loops and Continuous Improvement:**

Encourage frequent feedback from stakeholders, end-users, and team members. Use this feedback to adapt and improve the product continuously.

- **Empowering Teams and Individuals:**

Encourage teams to self-organize and make decisions. Trust their expertise and empower them to find the best ways to implement the functionalities assigned to them.

- **Value Stream Mapping:**

Map the value stream for each feature to identify bottlenecks, delays, or unnecessary steps in the development process and work on streamlining the workflow.

- **Visual Management and Information Radiators:**

Use visual tools like Kanban boards or Scrum boards to visualize the workflow, track progress, and make the project's status transparent to all stakeholders.

Aligning Lean Agile with Project Components:

- **Sprint Planning:**

Conduct sprint planning sessions at the beginning of each sprint to select and prioritize features from the backlog for that sprint.

- **Daily Stand-ups:**

Hold daily stand-up meetings to discuss progress, challenges, and plan for the day's work, ensuring everyone is aligned and aware of the project's status.

- **Sprint Reviews and Retrospectives:**

At the end of each sprint, conduct a review meeting with stakeholders to showcase completed features and gather feedback. Also, conduct a retrospective to reflect on the sprint and identify areas for improvement.

- **Minimum Viable Product (MVP):**

Identify and prioritize a Minimum Viable Product (MVP) that includes essential features to deliver value early to the users.

- **Kanban for Workflow Management:**

Use Kanban boards to visualize the workflow, manage tasks, and optimize the development process.

- **Continuous Integration and Deployment:**

Implement continuous integration and deployment practices to ensure that new features are integrated smoothly and deployed regularly.

By aligning the Lean Agile approach with the project components and using principles such as iterative development, feedback loops, and continuous improvement, you can efficiently deliver a high-quality product in a lean and agile manner, meeting the users' needs effectively.

Architectural Style:

1. Service-Oriented Architecture (SOA)

Description:

Service-Oriented Architecture (SOA) is an architectural style where components of the system are organized as services. These services are loosely coupled, meaning they can operate independently, yet can be integrated to achieve specific functionalities. Each service performs a specific business function and can be accessed and used independently by other components or services within the system.

Justification and Benefits:

- **Modularity and Reusability:**

- SOA allows breaking down the system into modular, independent services. Each feature in your project (e.g., authentication, chat, file compression) can be developed as a separate service.
- These services can be reused in future projects or scaled independently, promoting modularity and reusability.

- **Scalability:**

- Individual services can be scaled horizontally or vertically based on demand. For instance, if the chat service experiences heavy usage, only that specific service can be scaled without affecting other parts of the system.

- **Interoperability:**

- SOA enables interoperability between different services. Each service can be developed using different technologies or programming languages, as long as they communicate through standardized protocols such as HTTP/REST or SOAP.

- **Flexibility and Adaptability:**

- As services are loosely coupled, changes in one service do not impact other services. This flexibility allows for easier updates and modifications without disrupting the entire system.

- **Centralized Management:**

- SOA allows for centralized management of services. This means that each service can have its own development and deployment cycle, making it easier to manage the project as a whole.

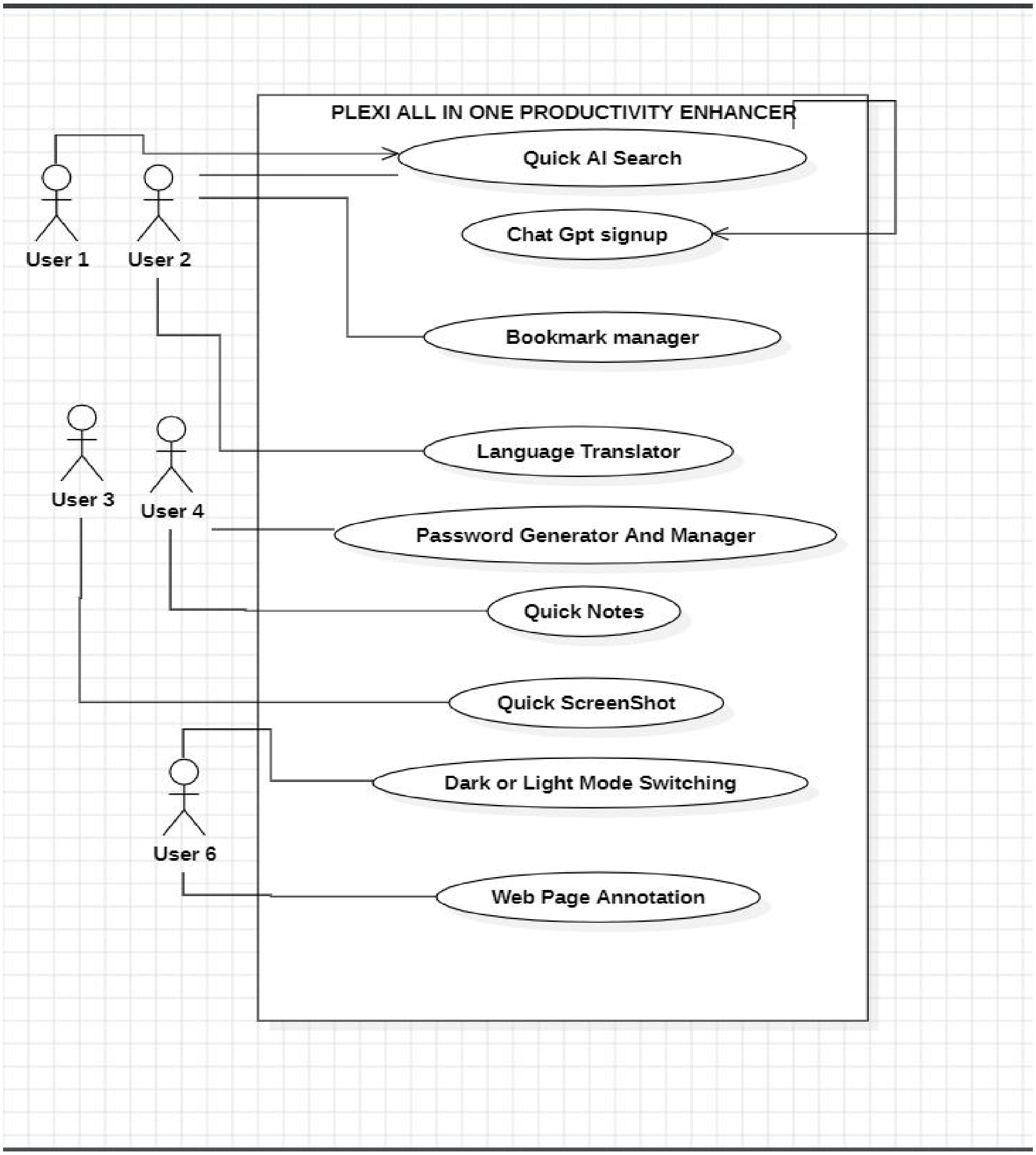
- **Enhanced Performance:**

- By breaking down complex tasks into smaller services, each service can be optimized individually for performance, leading to overall enhanced system performance.

- **Security and Access Control:**

- Security mechanisms can be implemented at the service level, ensuring controlled access to specific functionalities. This is crucial for features like user authentication and data privacy.

Considering the diverse functionalities of your project and the need for flexibility, scalability, and maintainability, a Service-Oriented Architecture (SOA) would provide the required structure and benefits to your system. Each component of your project, such as user authentication, chat, file compression, notes, and others, can be developed as individual services, enhancing the overall efficiency and maintainability of the system.



Test Plan, Cases

1. Introduction:

Purpose: The purpose of this test plan is to ensure the functionality, usability, and reliability of the browser extension.

2. Testing Scope:

In-Scope:

- Bookmark Manager
- Language Translator
- Quick Notes and To-Do Lists
- Adaptive Dark Mode
- Webpage Annotations
- Password Generator and Manager
- Quick Search AI
- Quick Screenshot

Out-of-Scope:

Features not mentioned in the project requirements.

3. Test Strategy:

-Testing Levels:

- Unit Testing
- Integration Testing
- System Testing

-Testing Types:

- Functional Testing
- Usability Testing
- Security Testing

4. Test Environment:

- Browser Versions: Chrome
- Operating Systems: Windows, Linux

5. Test Schedule:

-Test Milestones:

- Unit Testing: 20/10/2023
- Integration Testing: 25/10/2023
- System Testing: 05/11/2023

✧ Resource Allocation: Specify team members responsible for testing.

✧ Dependencies: Any external tools or services required for testing.

6. Test Deliverables:

List of test cases, test reports, and any other documents to be delivered.

7. Test Cases:

7.1 Bookmark Manager

Test Case UT_01: Creating a bookmark
Test Case Description: This test case verifies the functionality of creating a bookmark.
Pre-conditions: The extension is installed and a webpage is open.
Test Steps:

- Click on the bookmark icon.
- Enter a name and select a category.
- Save the bookmark.

Test Data: Name: "Example", Category: "Research"
Expected Results: The bookmark is created and visible in the "Research" category

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT_01	Bookmark Manager	Creating a bookmark	Extension installed, open webpage	1. Click on the bookmark icon. 2. Fill in name and category. 3. Save.	Name: "Example", Category: "Research"	Bookmark is created and visible in the "Research" category.	Bookmark is created and visible in the "Research" category.	Pass
IT_01	Bookmark Manager	Importing bookmarks	Extension installed, bookmarks exported	1. Click on the import button. 2. Select the exported file. 3. Confirm import.	Exported bookmarks file	Bookmarks from the file are imported successfully.	Bookmarks from the file are imported successfully.	Pass
ST_01	Bookmark Manager	Exporting bookmarks	Extension installed, bookmarks created	1. Click on the export button. 2. Save the file.		Bookmarks are saved in the exported file.	Bookmarks are saved in the exported file.	Pass

7.2 Language Translator

Test Case UT_02: Translating selected text
Test Case Description: This test case verifies the functionality of translating selected text.
Pre-conditions: The extension is installed and a webpage with text is open.
Test Steps:

- Select a portion of text.
- Click on the translate icon.

Test Data: Selected text: "Bonjour", Target Language: English
Expected Results: The selected text is translated to English.

Test	Name of	Test Case	Pre-	Test	Test Data	Expected	Actual	Test

Case ID	Module	Description	conditions	Steps		Results	Results	Result
UT_02	Language Translator	Translating selected text	Extension installed, open webpage with text	1. Select text. 2. Click on the translate icon.	Selected text: "Bonjour", Target Language: English	Text is translated to English	Text is translated to English.	Pass
IT_02	Language Translator	Translating entire webpage	Extension installed, open webpage	1. Click on the translate icon.	Target Language: Spanish	Entire webpage is translated to Spanish.	Entire webpage is translated to Spanish.	Pass
ST_02	Language Translator	Changing translation language	Extension installed, open webpage	1. Change target language. 2. Click on the translate icon.	Target Language: German	Entire webpage is translated to German.	Entire webpage is translated to German.	Pass

7.3 Quick Notes and To-Do Lists

Test Case UT_03: Creating a note
Test Case Description: This test case verifies the functionality of creating a note.
Pre-conditions: The extension is installed.
Test Steps:

- Click on the notes icon.
- Write a note.
- Save the note.

Test Data: Note: "Meeting at 3 PM"
Expected Results: The note is saved and visible in the list.

Test	Name of	Test Case	Pre-	Test	Test	Expected	Actual	Test

Case ID	Module	Description	conditions	Steps	Data	Results	Results	Result
UT_03	Quick Notes and To-Do Lists	Creating a note.	Extension installed.	1. Click on the notes icon. 2. Write a note. 3. Save.	Note: "Meeting at 3 PM"	Note is saved and visible in the list.	Note is saved and visible in the list.	Pass
IT_03	Quick Notes and To-Do Lists	Editing a task.	Extension installed, task created.	1. Click on the task. 2. Edit details. 3. Save.	Task: "Buy groceries"	Task details are updated.	Task details are updated.	Pass
ST_03	Quick Notes and To-Do Lists	Deleting a note.	Extension installed, note created.	1. Click on the delete icon. 2. Confirm deletion		Note is deleted and not visible in the list.	Note is deleted and not visible in the list.	Pass

7.4 Adaptive Dark Mode

Test Case UT_04: Enabling dark mode
Test Case Description: This test case verifies the functionality of enabling dark mode.
Pre-conditions: The extension is installed and a webpage without native dark mode support is open.
Test Steps:

- Click on the dark mode icon.

Test Data: N/A
Expected Results: Dark mode is enabled for the webpage.

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT_04	Adaptive Dark Mode	Enabling dark mode	Extension installed, open webpage without native dark mode support.	1. Click on the dark mode icon.		Dark mode is enabled for the webpage.	Dark mode is enabled for the webpage.	Pass
IT_04	Adaptive Dark Mode	Disabling dark mode	Extension installed, open webpage with dark mode enabled .	1. Click on the dark mode icon.		Dark mode is disabled for the webpage.	Dark mode is disabled for the webpage.	Pass
ST_04	Adaptive Dark Mode	Configuring dark mode settings	Extension installed.	1. Open extension settings. 2. Adjust dark mode settings. 3. Save.	Dark mode settings: Contrast: High, Opacity: 70%	Dark mode is customized as per settings.	Dark mode is customized as per settings.	Pass

7.5 Webpage Annotations

Test Case UT_05: Highlighting content
Test Case Description: This test case verifies the functionality of highlighting content on a webpage.
Pre-conditions: The extension is installed and a webpage is open.
Test Steps:

- Select content on the webpage.
- Click on the highlight icon.

Test Data: Selected content
Expected Results: The selected content on the webpage is highlighted.

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT_05	Webpage Annotations.	Highlighting content.	Extension installed, open webpage	1. Select content. 2. Click on the highlight icon.	Selected content	Content is highlighted.	Content is highlighted.	Pass
IT_05	Webpage Annotations.	Adding a note.	Extension installed, open webpage	1. Click on the note icon. 2. Write a note. 3. Save.	Note: "Important point"	Note is saved and associated with the webpage.	Note is saved and associated with the webpage.	Pass
ST_05	Webpage Annotations.	Exporting annotations.	Extension installed, annotations created	1. Click on the export icon. 2. Save the file.		Annotations are saved in the exported file.	Annotations are saved in the exported file.	Pass

7.6 Password Generator and Manager

Test Case UT_06: Generating a password
Test Case Description: This test case verifies the functionality of generating a password.
Pre-conditions: The extension is installed.
Test Steps:

- Click on the password icon.
- Configure options for the password.
- Generate the password.

Test Data: Length: 12, Include Symbols: Yes
Expected Results: A password is generated according to the specified options.

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT_06	Password Generator and Manager	Generating a password	Extension installed	1. Click on the password icon. 2. Configure options. 3. Generate.	Length: 12, Include Symbols: Yes	Password is generated as per the specified options.	Password is generated as per the specified options.	Pass
IT_06	Password Generator and Manager	Saving a password	Extension installed, generated password	1. Click on the save icon. 2. Provide a name and category. 3. Save.	Name: "Email Password", Category: "Work"	Password is saved and visible in the "Work" category.	Password is saved and visible in the "Work" category.	Pass
ST_06	Password Generator and Manager	Copying a password	Extension installed, password saved	1. Click on the copy icon next to the password.		Password is copied to the clipboard.	Password is copied to the clipboard.	Pass

7.7 Quick Search AI

Test Case UT_07: Conducting a quick web search
Test Case Description: This test case verifies the functionality of conducting a quick web search using AI.
Pre-conditions: The extension is installed.
Test Steps:

- Enter a search query in the search bar.
- Press Enter.

Test Data: Search Query: "Productivity tips"
Expected Results: Relevant search results are displayed in the extension.

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT_07	Quick Search AI	Conducting a quick web search	Extension installed	1. Enter a search query in the search bar. 2. Press Enter.	Search Query: "Productivity tips"	Relevant search results are displayed in the extension.	Relevant search results are displayed in the extension.	Pass
IT_07	Quick Search AI	Refining search results	Extension installed, search results displayed	1. Apply filters or refine search query.	Filters: Time: Past Month, Source: Blogs	Search results are updated based on applied filters.	Search results are updated based on applied filters.	Pass
ST_07	Quick Search AI	Opening search results	Extension installed, search results displayed	1. Click on a search result.		Selected search result is opened in a new tab.	Selected search result is opened in a new tab.	Pass

7.8 Quick Screenshot

Test Case UT_08: Capturing a screenshot
Test Case Description: This test case verifies the functionality of capturing a screenshot of a webpage.
Pre-conditions: The extension is installed and a webpage is open.
Test Steps:

- Click on the screenshot icon.
- Select an area to capture.

Test Data: Selected Area
Expected Results: A screenshot is captured and displayed in the extension.

Test Case ID	**Name of Module**	**Test Case Description**	**Pre-conditions**	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Test Result**
UT_08	Quick Screenshot	Capturing a screenshot	Extension installed, open webpage.	1. Click on the screenshot icon. 2. Select area to capture.	Selected Area	Screenshot is captured and displayed in the extension.	Screenshot is captured and displayed in the extension.	Pass
IT_08	Quick Screenshot	Annotating a screenshot	Extension installed, screenshot captured.	1. Click on the edit icon. 2. Add annotations. 3. Save.	Annotations: "Note on screenshot"	Annotations are added to the screenshot.	Annotations are added to the screenshot.	Pass
ST_08	Quick Screenshot	Managing screenshots	Extension installed, multiple screenshots captured.	1. Click on the screenshot icon. 2. View the screenshot history.		List of captured screenshots is displayed.	List of captured screenshots is displayed.	Pass

Conclusion

The above test cases cover the functional features of the All-in-One Productivity Chrome Extension, including Bookmark Manager, Language Translator, Quick Notes and To-Do Lists, Adaptive Dark Mode, Webpage Annotations, Password Generator and Manager, Quick Search AI, and Quick Screenshot tool. Each test case is categorized into Unit, Integration, and System testing to ensure comprehensive coverage.