Project Title: Chrome Extension - All-in-One Productivity Enhancer

Team Number : 5

- Sarthak S Kumar (SRN: PES2UG21CS482) -

  Sathish Kumar G (SRN: PES2UG21CS484)

- Sanath Kumar R (SRN: PES2UG21CS474)

- Sumukha N M (SRN: PES2UG22CS822)

SDLC Model

1>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>For a software development project with the mentioned features and functionalities, here are the recommended tools for each phase of the 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.

In your software development project, there are several deliverables that can be categorized as either reusable components (components that can be used in multiple parts of the system or in future projects) or build components (specific to this project and not likely to be reused). Here's a breakdown:

Reusable Components:

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.

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. *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.

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.

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.

6. *Text-to-Speech Integration Module:*

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

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.

8. *Password Saving Interface and Integration:*

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

9. *Translation API Integration:*

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

10. *Calculator Interface and Integration:*

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

Build Components:

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.

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. *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.

4>Work breakdown structure

<u>Plan of Work and Product Ownership:</u>

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

<u>Proposed Project Description:</u>

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, Webpage Annotations, Password Generator and Manager, Quick Search AI, and a Quick Screenshot tool.

<u>Functional Features and Qualitative Properties:</u>

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.

- Functional Features: Adaptive Dark Mode, Web annotations

5>**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

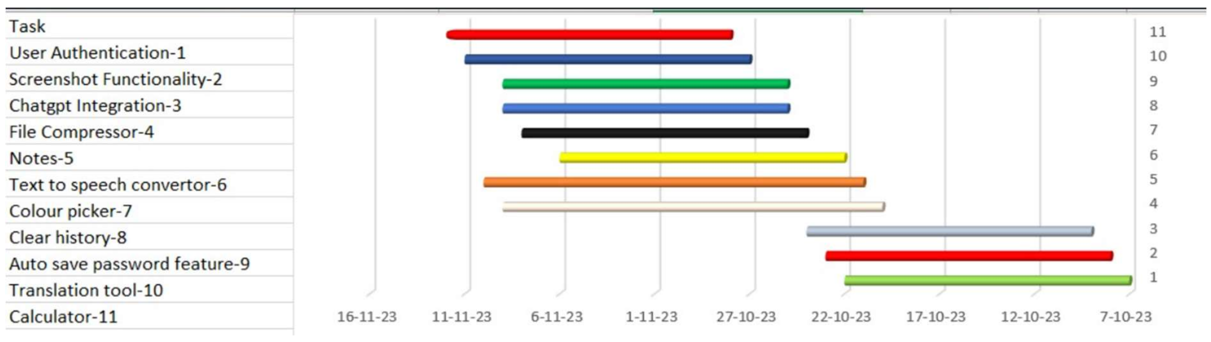
- Integrating calculator functionality with history using off-the-shelf API

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.

6>Gantt chart visualization


Task
User Authentication-1
Screenshot Functionality-2
Chatgpt Integration-3
File Compressor-4
Notes-5
Text to speech convertor-6
Colour picker-7
Clear history-8
Auto save password feature-9
Translation tool-10
Calculator-11