Link to the current project: [Scholarly Current Active Webtool](https://scholarlybook.com/)

**Project stack (NOTE: We can change it to Linux by upgrading tool all over)**

* .Net
* MySQL database
* Angular.js

**Software's integrated**

* annotator.js

**Hosting**

* AWS windows server

**Goal:** By 31st December 2024, update the current scholarly book project by enhancing UI/UX design and the backend with integrating secure GPT models (free or paid), plagiarism detection software (free or paid), and a legal payment gateways to improve article reading and reviewing experiences, catering to both individual subscribers and institutional clients.

**Timeline:** 3 months

**Budget:**

**Stack:** Instead of sticking to same stack we can change to other stack that would support the GPT model or other software integration and avoid dependance on Windows licensing.

**Specific Requirements:**

NOTE: These are broader thoughts, but we can refine them based on easy to implement and long term goals.

**UI/UX Design**

* Cleaner, responsive with clear message about the tool
* Accessibility
* Responsive Design

**After Login**

* Dashboard Customization
* Encourage user engagement and retention.

**AI Integration for user reading an article**

* **Out-of-Box AI based Q&A on the uploaded article**: When a user uploads a PDF article, a new tab with `AI assistant` should have a question and answer of the article, which broadly helps to understand the complete article at a glance. Eg: questions below. ( NOTE: open for a simpler and better approach)
  + Let user to accept or reject or edit it to reframe the question, a way of capturing his feedback on the AI , so that we can train the models in future use
* **Contextual Understanding**: When user highlights a content on the PDF article, suggest phrasing questions using AI
  + Capture the user annotations in the database, so that we can integrate semantic search models in future or train models so that AI can automatically ask article relevant questions instead of the below set of questions we begin with.
* **Example questions** : **NOTE**: Some are direct, some are analytical, if AI models out-of-box can capture the direct context questions, we would like to implement that first and go live.
  1. What is the main objective of the research?
  2. What methodology was used in the study?
  3. What are the key findings or results?
  4. What conclusions did the authors draw?
  5. What are the limitations of the study?
  6. What motivated the researchers to conduct this study?
  7. How does this research relate to existing literature?
  8. What are the practical implications of the findings?
  9. What assumptions underlie the research?
  10. What statistical methods were used to analyze the data?
  11. How robust are the study’s findings?
  12. Are there any potential biases in the study design or data collection?
  13. How do the results compare with previous studies on the same topic?
  14. What are the potential future applications of this research?
  15. How could this research be expanded or built upon in future studies?
  16. What new questions have emerged as a result of this study?

Eg: The expected output is something like below. If there is a relevant content regarding any of the question above on that article uploaded, we would like to either highlight in the PDF, and let the user to hover to see what the highlight means

A screenshot of a message

Description automatically generated

And a tab ( AI assistant) where these are pulled in the format as Q&A .

Eg: Below. NOTE: The questions are not relevant to those 16 written above, but the AI assistant tab should have something like this. If there is no relevant data to the question, just leave it as blank or NA. A screenshot of a phone

Description automatically generated

**Features for reviewing article for un-published works**

* **NLP based Analytics**: Grammarly score, or readability scores, or duplicate or repetitive sentences identification or plagiarized paragraphs identification or Language Assessment tool score (<https://www.aje.com/grammar-check/> )
* The goal is here to just a new feature for Scholarly users , so that they can review articles more accurately, and gives a basic assessment even before reading the article
* We do not need to capture the user annotations, as this is unpublished work

We are looking for analytics something like this (NOTE: representative images)

A screenshot of a computer

Description automatically generated

**A screenshot of a cell phone

Description automatically generated**

**Payment System Integration**

* **Multiple Payment Options**: Offer multiple payment options, including credit/debit cards, and other institutional payment methods.
* **Subscription Management**: Provide users with tools to manage their subscriptions, including upgrading, downgrading, and canceling subscriptions.
* **Invoicing**: Generate and send invoices automatically for institutional clients.
* Representative images below, we can come with more exact pricing once we assess the AI cost, hosting cost and development cost

A screenshot of a website

Description automatically generated

Source: Competitor <https://www.read.enago.com/pricing/>

**Single Sign-On Integration**

* **Easy sign in options**: Support users to log in using their institutional credentials or can do a quick sing-up
* **Data Privacy**: Ensure that user data is handled in compliance with relevant data privacy regulations and encryptions

**Feedback and continuous development**

* **Feedback Loop**: Create a feedback loop where users can provide feedback using a quick chat box or directly on the on the AI-generated content and other features, helping to continuously improve the system.
* **API Integration**: Develop application in such a way that we can implement APIs for future developments

**Security Enhancements**

* **Data Encryption**: Ensure all data, both in transit and at rest, is encrypted using industry-standard encryption protocols.
* **User Authentication**: Implement multi-factor authentication (MFA) to enhance user account security.
* **User Control privacy setting**: Let user control data sharing

**Phase -1**

**Let’s plan for first phase in 3 months**

* 1. Convert the current site with better UI and that can run in Linux server
  2. Add the current functionality
     + Annotator.js is integrated ( annotations)
     + We create groups
     + Choose privacy setting
     + Download selected annotation
     + Feedback and user-sign ( email notifications)
  3. Document review score
  4. Payment gateway

The goal is to re-brand as an AI powered. The current Scholarly is built with an idea as a demo tool, it is attracting users to sign-in, but does not retain users. The goal of phase 1 is to re-brand Scholarly, to engage users and be able to monetize, for more focused AI implementation in future.

Breaking down to a task level to discuss the possibility of implementation 3-month time frame.

**Infrastructure**

* Using Linux server instead of windows server
* Appropriate database for goals, mix of MySQL, graphdB or other open source
* Update annotation.js or using pubtator as the annotation engine
* Decision on Stack for development taking cues from existing competitors

**UI/UX**

1. Design appealing Front-end like competitors
   1. Easy sign-in approach
   2. Necessary tabs
   3. Legal terms and conditions
   4. Limited Animations demoing how to use

**Sign-in options**

1. How should users sign-up ?
   1. Basic email sign-up
   2. Ouauth ( google, twitter)
   3. Edu or institutional for bulk accounts

**Payment options**

1. Similar to competitors
   1. Free tier with limited paper upload and annotation capability
   2. Different tiers with respective payment integration ( credit/debit)

**After Logging in**

1. User level controls
   1. Basic Form to fill details of user’s interests
   2. Control given to user on what level security he would like to implement on his details
   3. Ability to create multiple groups and the groups privacy controls
   4. Dashboard organizing or other UI elements user make like to control
2. Ingesting a research article
   1. A screen like Pubtator for user to search article while logging in Scholarly
   2. A Scholarly plug-in that user can install in browser, so that user can add an article from google search while signed in the plug-in
   3. Upload a PDF file
   4. Every ingested article will have a structured information about the article that we can capture in a form , user should be able to edit or add if the article ingestion failed to capture that info (reference managers do it already)
   5. Ability to organize the ingested articles in folders, that users have control
   6. Ability to output as different citation styles
3. Annotating a research article
   1. Make the reading experience more relaxing on eye
   2. Select and right click to add annotations, color code them, add tags
   3. Like, Google Dictionary , ability to double click a word to get its meaning, this dictionary would have more scientific terms.
   4. We need to address user purpose of reading and ability to full-fill that purpose
   5. Options to export all or certain annotation along with its citations to csv, google docs or word document
   6. Out of the box AI suggested question or annotation framing
   7. A feature that would help user to organize notes and thoughts
4. Article Review Score
   1. Allow user to upload a to-be published article
   2. Provide basic analysis dashboard on document
   3. Provide a score on the document, plug-in
5. Google docs tab
   1. Provide an ability to the user to select content from article reading tab and send it to the google docs along with the article citation
   2. Provide template options for user ( an AI out of the box solution) 
      1. Eg: Provide a template outline for writing a review article
6. User personal dashboard
   1. To see analytics on the articles read 
      1. Eg: Some bubble charts for his tags, or number of annotations he did, articles he read etc. Something informative
7. Explore competition websites and implement any simple integrations which does not require extensive development such as…
   1. Right click to search wiki, google
   2. Color based segmentation of article