CrowdLib Proposal

## **Team**: Team Blue

**Members**: Joshua Kibreab, Andy Payan, Max Phung, Shane Zimmerman

**Project Motivation**:

Our goal is to bring the classic word game, Madlibs, to the internet. In the form of a web application, we want to create a space for users to share their experiences with Madlibs to offer a community. Traditionally, Madlibs are usually performed in person on a piece of paper and shared with a small group of people. We will implement features where users can create an account, post, vote, and comment. One large feature is AI image generation that will be used to offer a visual to a user’s Madlib creation. Instead of having to use one’s imagination, their creations will be illustrated and be created into a real story.

**Functional Requirements**:

**Platform**: Web Application

**Front-End**: JavaScript, React

**Back-End**: Python, Django

**Database**: MongoDB, AWS S3

**Main functionality:**

1. **Account Creation:** I should be able to make an account using sign-in partners like Google and Github. These accounts should store my madlibs, social interactions, and settings so that they are available every time I log in.
2. **Fill in, Save, and Share Madlibs:** I should have the option to select from many different madlibs. After I fill in a madlib, I should be able to access it again and share it with others on and off the platform.

1. **Follow Other Users:** If I see a madlib that I really like, I should be able to follow the author so that I can see more of their work in the future. I want a feed that shows me the public madlibs created by the users I follow.
2. **Posting/Commenting/Liking Feature**: I want to be able to either keep my madlib private, or post it for others to see. I should be able to like and comment on madlibs. I should also be able to like and reply to other users' comments.
3. **AI Madlib Image Generator:** When I finish filling in a madlib, I should have the option to generate an AI image based on the prompt. I want to also be able to choose the style of the image generated.
4. **Filter and Browse the best Madlibs:** I want a feed that allows me to see the best madlibs from the community. I should be able to filter and refine this feed based on the number of likes or comments.

**User Cases:**

#### **User Case 1: Account Creation & Login**

**Actors:** New User, Returning User  
**Goal:** Allow users to create an account and log in securely to access the website  
**Precondition:** The user has internet access and opens the CrowdLib site.

**Main Flow:**

1. The user clicks on “Sign Up” or “Login.”
2. The user selects Google OAuth 2.0 or enters other required credentials.
3. Upon successful authentication, the user is redirected to the home feed.

**Postcondition:** The user’s account is created (if new) or their session is authenticated (if returning), allowing access to all platform features.

#### 

#### **User Case 2: Create and Fill Out a Madlib**

**Actors:** Registered User  
**Goal:** Enable users to create or fill in a Madlib story template.  
**Precondition:** The user is logged in.

**Main Flow:**

1. The user clicks “Create Madlib” or selects an existing template.
2. The user fills in the blank fields with words of their choice.
3. The completed Madlib story is displayed.
4. The user can choose to save it as a draft or post it publicly.

**Postcondition:** The Madlib is stored in the database, either as a private draft or as a published post.

#### **User Case 3: AI image generation for existing Madlibs**

**Actors:** Registered User  
**Goal:** Allows user to generate an image that represents their Madlib  
**Precondition:** The user has created or selected a pre-existing Madlib.

**Main Flow:**

1. The user selects a Madlib, either a pre-existing one or a newly created one.
2. The user clicks “Generate Image”
3. The AI image generator grabs the Madlib prompt and uses it to generate a corresponding image.
4. The image is displayed alongside the original Madlib entry.

**Postcondition:** The image is saved to the project’s AWS S3 bucket and a reference link to the image is stored in the database. The images are either to the user or can be shared publicly.

#### **User Case 4: Browse and interact with Shared Madlibs & Images**

**Actors:** Registered User  
**Goal:** Allows users to explore various Madlibs and AI-generated images other users have already created. Also allows users to interact with posted content with likes and comments.  
**Precondition:** The user is logged in.

**Main Flow:**

* 1. The user scrolls through the feed or filters by “Most Liked” or “Trending.”
  2. The user can view Madlibs along with their generated images.
  3. The user can like, comment, or share posts.

**Postcondition:** User interactions are saved in the database

**Feasibility Analysis and Challenges:**

This project is feasible within allotted time because of our diverse technical backgrounds. With the 8 weeks remaining in the quarter, we should be able to fully develop our web app and have it be fully functional. Everyone in Team Blue has some sort of experience with programming languages such as C++, python, javascript, etc..

Some members have experience with more diverse applications and languages that will help move this project off the ground. Shane has experience with MongoDB along with Django to help with our backend needs. Josh has beginner experience with Django as well. Max and Andy both have general experience with web development, more specifically front-end tasks like visuals, responsiveness, etc.

We will organize our time for developing the website using a Kanban board, where each member will have specific tasks assigned to them. The teammates with more backend experience will focus on more backend tasks, while the ones with more frontend experience will focus on the tasks at front. Given our skill set, resources, and clear organization, completing this project within the quarter is realistic.