

## Digital Library - Week 1 Report

By: Joshua Watson, Camron Mellott, and Luke Joseph

The Digital Library project was officially started this past week. The vision for Digital Library is a web and mobile application that allows users to add books manually, scan ISBN codes to add books, and gain AI recommendations. To get started on the project, we needed to set up our project environment and install the necessary applications and frameworks.

Firstly, we needed a version control software. Working as a team, being intentional about version control will be imperative to the success of the project. We chose Git as our version control software, which is the most popular and well-known software on the market. Once we had a means of staying on the same page within the project, we were ready to start installing the necessary applications and runtime environment. The first thing we needed to do was get the runtime environment installed. Since we are creating a web application, we determined JavaScript to be the best language for the project. JavaScript doesn't natively support web applications so we had to install Node.js, a runtime environment that gives JavaScript the capabilities to create web servers and applications. Once we had Node.js installed, we then needed NVM and NPM. NVM stands for Node Version Manager, which allows for quick and easy switching between versions of Node. NPM stands for Node Package Manager, which will manage all the installations of dependencies we'll be using throughout the project.

Afterwards, we had the basis for our project and were ready to start setting up the environment we will be working in from now on. The Digital Library is planned to be both a web and mobile application, to fulfill this goal, we have decided to use Cursor, which will serve

as both a mobile wrapper for our application and the IDE we will use to code the project. Finally, we were ready to set up our project structure. We created a Github repository, and cloned it across all machines. Once complete, we created a new branch to begin work on the UI.

This initial week, we faced several challenges. The first problem that we faced had to do with our lack of knowledge of JavaScript. None of us has more than a vague idea of how to write in JavaScript, leading to the initial problem of not knowing how to set up our environment. Another problem was the differing operating systems for each member. Two of our members use Unix based machines, and one of our members uses a Windows based machine. This made every step of initialization different for our team. Finally, one issue we've faced is with the ISBN scanner API. We have looked into the APIs available for our barcode scanner and have found that there aren't any open source APIs. In addition, the APIs that are available charge \$99 a month for calls.

We were able to solve two out of the three problems we faced last week. For the JavaScript issue, we simply had to look into the Node.js documentation. When it came to learning about NVM and NPM, Google's AI, helped out a lot, providing context and even terminal commands that helped move along the installation processes. For the operating system problem, again, Google AI helped out a lot with installation commands and overall setup, specifically for our Windows user. The only problem we were unable to solve is the scanner API problem. We still do not know if we are going to have to pay for this specific API or if we can find a free and reliable one elsewhere.

Now, we are in line with our Gantt chart progress, which has us in line with our February 6<sup>th</sup> deadline for the UI. The plan for this week is to work on the basic UI features, such as the

search bar and basic menus. We worked collaboratively on each part of both the project and report, for each member's contributions:

Joshua Watson: Report, initialization

Luke Joseph: Slides, initialization

Camron Mellott: Slides, initialization