**8WOC 2018 Backend Coding Challenge**

**Challenge**: A Front-End Developer at Wycliffe Associates developed the front-end for a web application. This application prompts the user to type in the book title, chapter number, and verse number for a passage in the Bible. Once entered, the user must click a button, and that specific passage will pop-up on the user’s screen in an editable text field. However, the Front-End Developer has little experience developing backends for applications. The developer needs your help creating an API that integrates with the front-end code found at the following URL: <https://github.com/Pinto18/frontend_8woc2018> .

**Criteria:** Clone the GitHub repository found at the URL above, and create an API which provides endpoints for the application so that we can develop a fully-functioning Bible passage querying web app. Feel free to use any programming language you want to develop this API. This project will require setting up a database to query from that matches the information the user must provide to the UI. Any query to your database should return a JSON object of the following format:

{

“book” : “John”,

“chapter\_num” : “3”,

“verse\_num” : “16”,

“verse” : “hello”

}

Currently, the front end is configured to run on <http://localhost:3000>. Feel free to make any necessary changes to the front-end code to improve the configurations or UI. Use the user stories below to get a better understanding of what features we are asking you to code:

**User Story:**

1. As a user, I want to search for a verse in the Bible so that I can recall that specific verse.

**Extra Credit:**

**\*PLEASE NOTE:** The criteria below are NOT mandatory. The purpose of these extra credit criteria is to help us to get a better sense of your skills and personality. Failure to complete one or more of these criteria will not count against you in our evaluation of your submission.

1. Create a GitHub repository of your final project and send us a link to it as a submission. We are looking for use of branches, and a history of pull requests, etc.
2. Develop and execute unit tests which verify that the API works as expected. Include in your submission some form of unit test report.
3. Include functionality to edit verses and save changes.
4. Include functionality to add new verses using the front end.
5. Attempt to enhance the UI of the web app
6. Configure the app to run on your own server which we can connect to