Web & Mobile Programming
Semester – Summer 2020

SHIFU
(MEAN Stack Project)

Team 7 - Members

Yamini Reddy Dontireddy(16293335)
Satya Anusha Prattipati (16292575)
Aarthi Nagireddy(16292448)
Sandeep Meesala(16299481)

Contents

Objective

The main objective of our project is to demonstrate the capabilities of ***Microservices Architecture***, especially developing simple application or use case in **MEAN** stack with all front end, service layer and a data store.

Shifu aka Master Shifu, a secure online application that serves as open online education platform for the community.

Terminology

MEAN – **M**ongoDB, **E**xpress **F**ramework, **A**ngular and **N**ode JS
MOOC – **M**assive **O**pen **O**nline **C**ourses like CourseEra or MIT OpenCourseware
Rest API – **Re**presentational **S**tate **T**ransfer
JSON - JavaScript **O**bject **N**otation
BSON – **B**inary **JSON**, format used in MongoDB
API – **A**pplication **P**rogramming **I**nterface
CRUD – **C**reate, **R**ead, **U**pdate, **D**elete
Endpoint - An endpoint describes the ReSTful interface for a service implementation.

Motivation

Shifu aka Master Shifu, Envisioned as an open community platform where scholars can directly schedule their seminars or teaching sessions and students communities can choose their topic of interest and attend those sessions.

“Students Without Borders & Teachers Without Borders”

Significance

Similar to Khan Academy or any MOOC, this platform has large role TO play in educating the masses who need affordable education within their limits. Unlike Khan Academy, this platform is conceptualized as crowd sourced, crow funded and patronized by reputed scholars from society to deliver the sessions.

This platform would enable professors and students across the globe to collaborate and drive education in all native languages that can be accessible across the globe.

Recordings would also be available for offline education and multiple other features to enable full class room environment would be provided in future phases.

Quality of the content and faculty would be regulated through direct feedback of users and also through quality reviews through peer groups.

Features

Current phase is a pilot implementation with simple use case with below features,

- Courseware creation by Faculty users
- Edit / Delete Courseware by Faculty users
- Search Courseware by Student users
- Schedule Sessions by Student users
- Resources with course video library (Completed)
- Integration with Google Library for books
- Google Location API integration

Tool & Technologies

Front End

- Angular
- HTML
- CSS
- TypeScript
- JavaScript

Back End

- MongoDB

Services

- Nodejs

Framework

- Express

Code Management

- GitHub
- WebStorm

Architecture

Microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment process. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies. “Martin Fowler”

MEAN stack is chosen as the reference architecture for this project and it provides modular architecture with coupled services which can be developed, deployed and work independently.

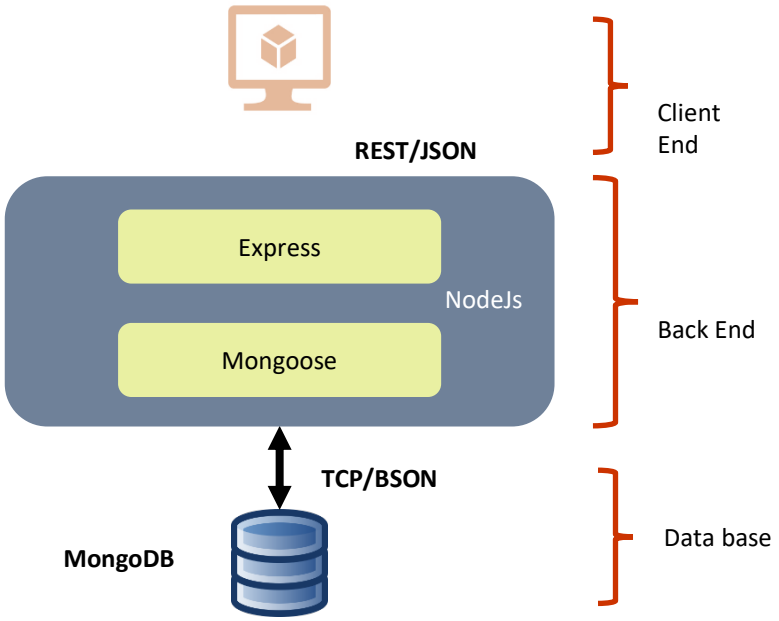
M **MongoDB**, the world's leading NoSQL database that stores its data into a JSON-like formatted binary file (Binary JSON - BSON). Easy to use and simple to work with JSON.

E **Express**, a lightweight, minimalist framework built for Node.js for creating web applications, APIs, and HTTP facilities.

A **AngularJS**, the Model-View-Whatever JS framework makes API consuming simple, optimized for mobile development, same code base for browser, etc.

N **Node.js,** runs on Chrome's V8 engine and capable of non-blocking, event-driven I/O, handle multiple requests on a single service without them blocking each-other (hence non-blocking).

Shifu Architecture



API

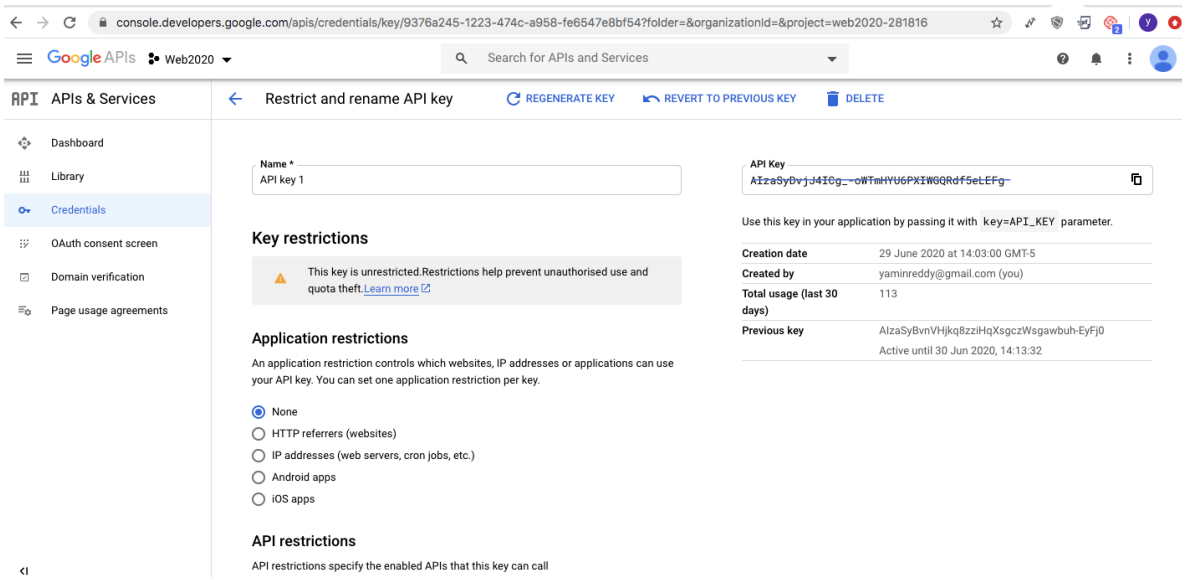
Two APIs are used to integrate with Geo Location and Google Books library.

- **Google Maps and Places API**
- **Google Books API**

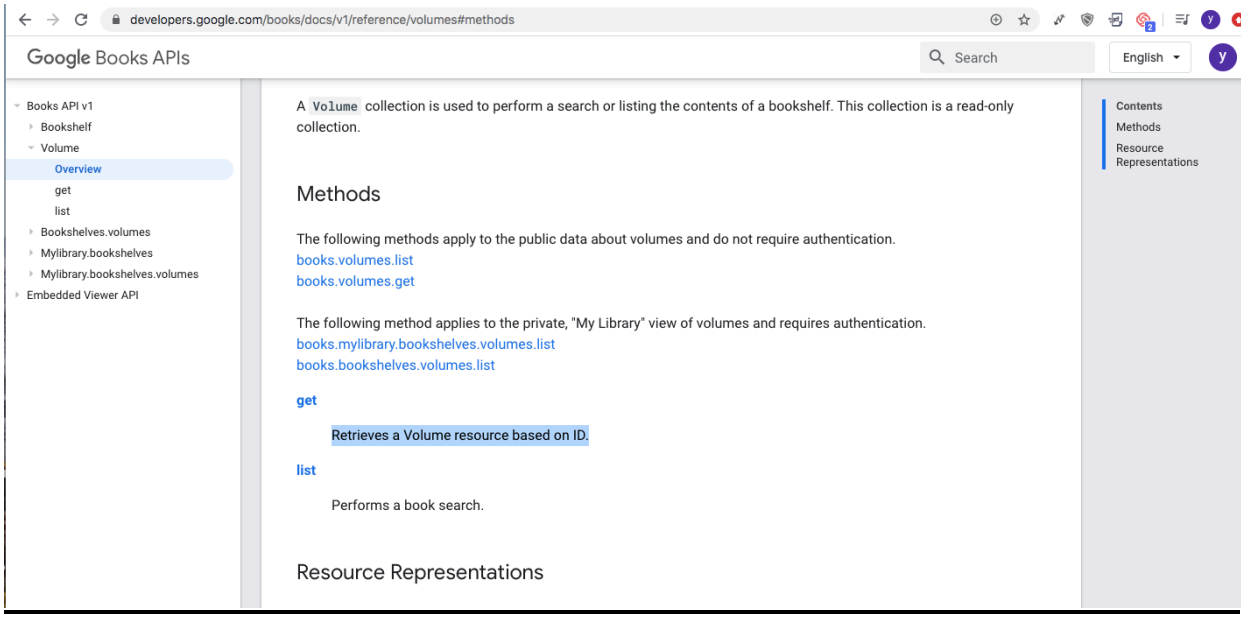
Google exposes APIs across all its applications to the community and any developer can sign up for Developer account to consume or use these APIs. Singed up in google developer account and got the End Points and API secret Keys.

Shifu is integrated with both location and map for students to choose accurate location and find those locations in Google map. Students can also search for the book's (integrated with Google Books API).

Places and Maps API



Books API



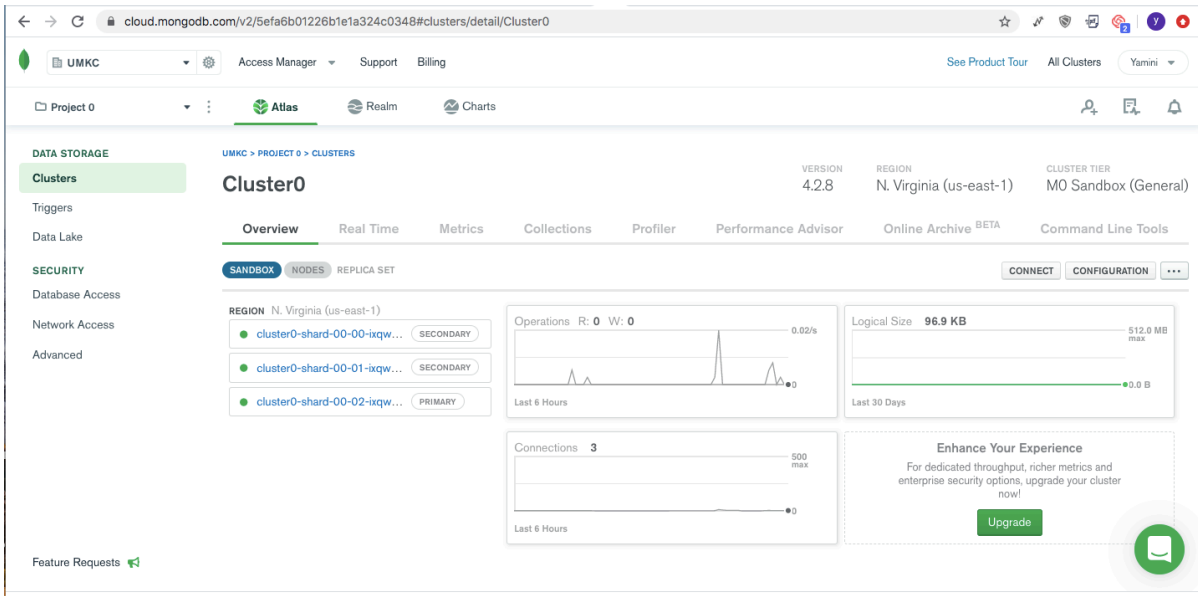
Data Store

MongoDB Cloud

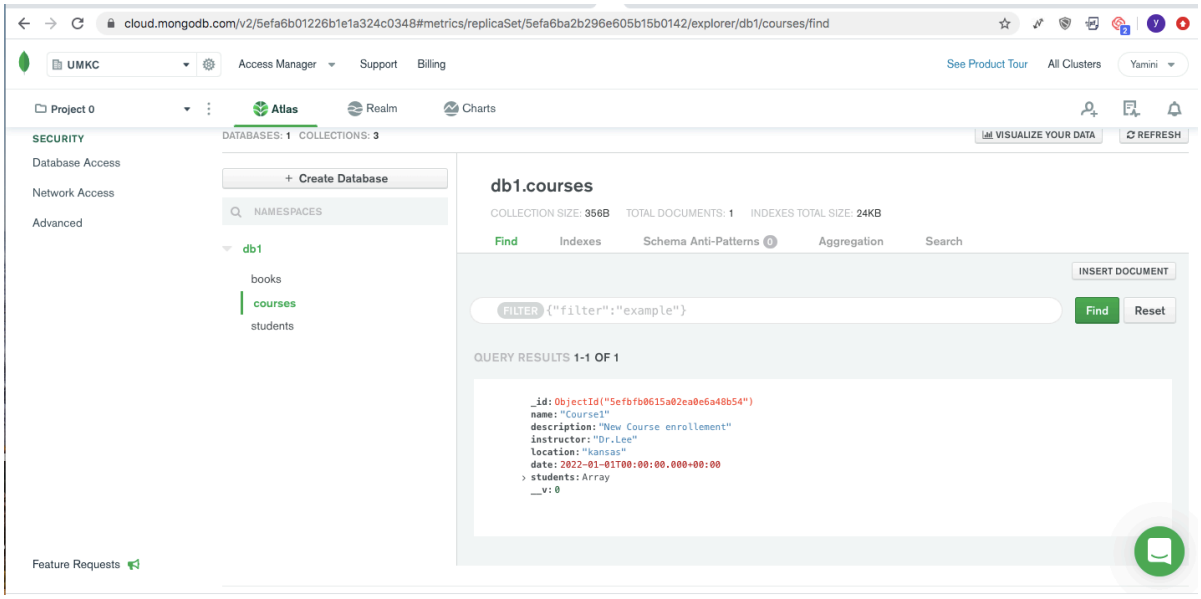
MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

The application developed is connected to the cloud DB through Express framework. MongoDB stores BSON documents, i.e. data records, in collections; the collections in databases.

Below are the connection and the DB details,



Mongo DB Collections

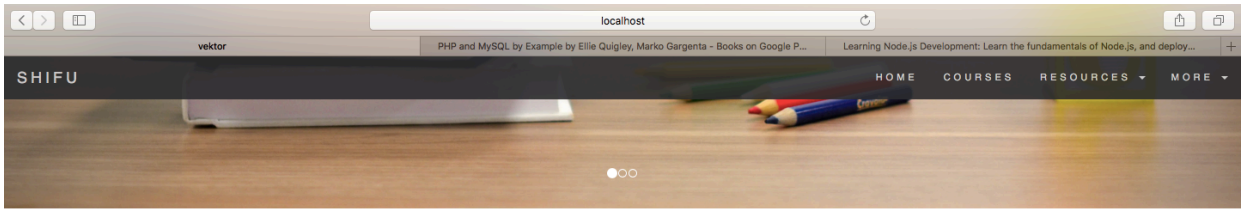


Application

Application is developed using Angular to align with overall architecture and responsive UI that is adaptable to all screens.

Front End Overview – Application contains below screens for the ease of use and designed intuitively based on cognitive needs.

Landing Page – Slider screens with colorful UI to easily connect with user and convey the mission of this platform.



About SHIFU

Course Registration & Management

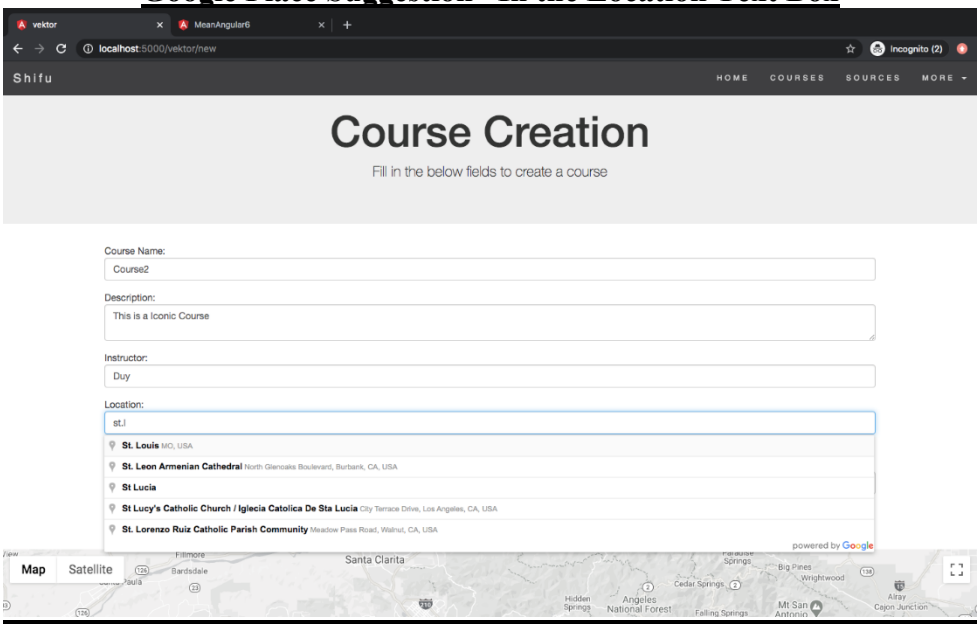
Shifu provides Course creation and registration solutions for any training centers, corporate and workforce training organizations. This Platform makes it easy for students to register for the classes with a simple enrollment form. Securely collects the student information and course selection data to make the online course enrollment a smooth and efficient process.

Google Maps and Books API are also integrated in this application.

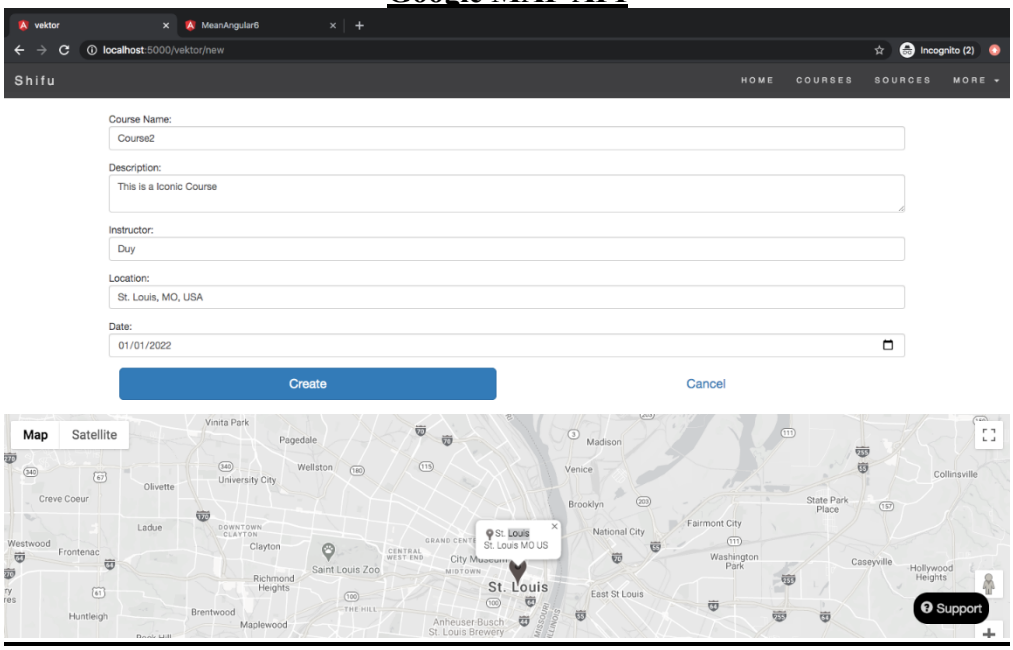
<u>Courses Creation Page:</u> Faculty users can log on to the platform and design courses based on their expertise. Role based access for students and faculty is planned for future development.

Core creation involves a CRUD operation that invokes node service to store the course information in data store vis JSON.
Also integrated with Google Maps and Places API to enable location suggestions in the location field to capture accurate location information. Once the location is entered, the user can view the map of the location below (Where the Google MAP API is called)

<u>Google Place Suggestion– In the Location Text Box</u>

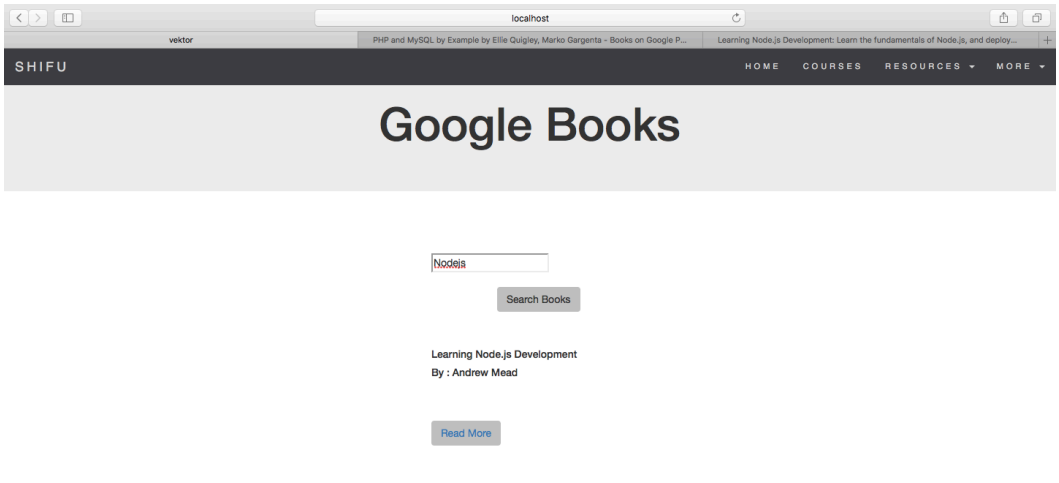


Google MAP API



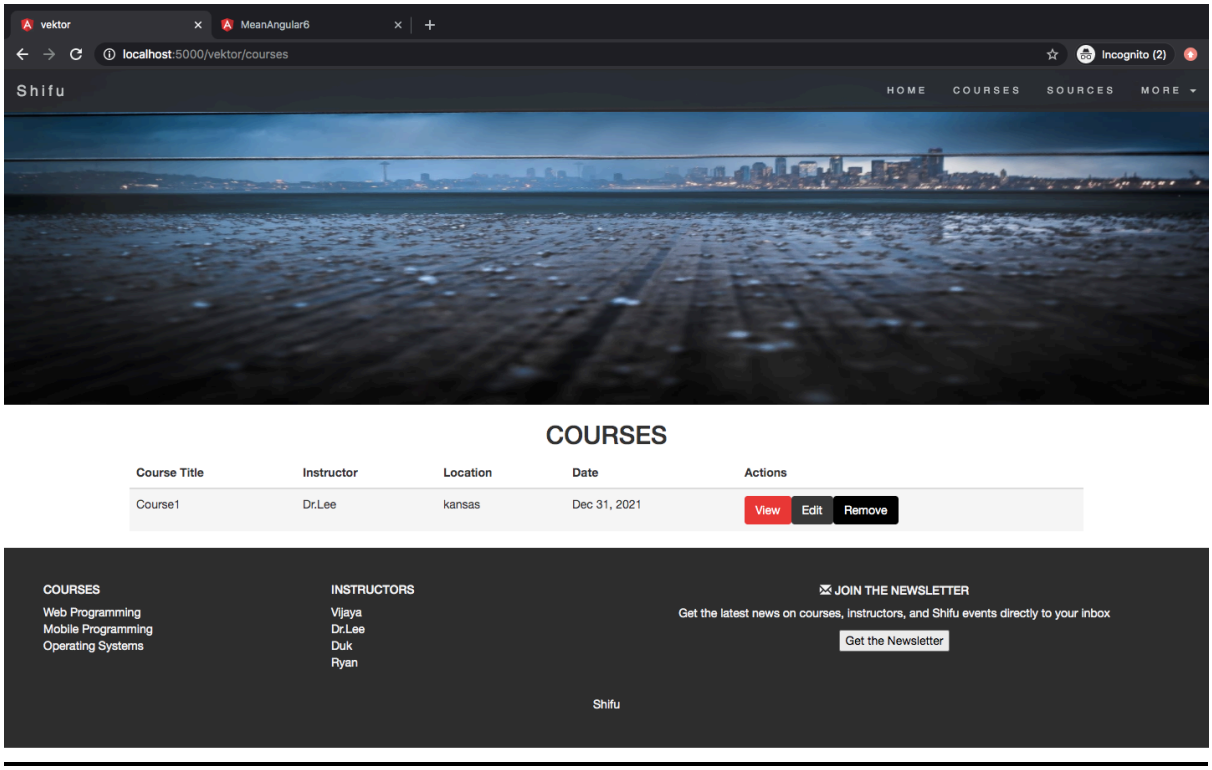
Google Books Page: Integrated with Google Books API ,the user can click on the "Read More" button and can navigate to respective site (Where the Google Books API is called) .

Google Books API



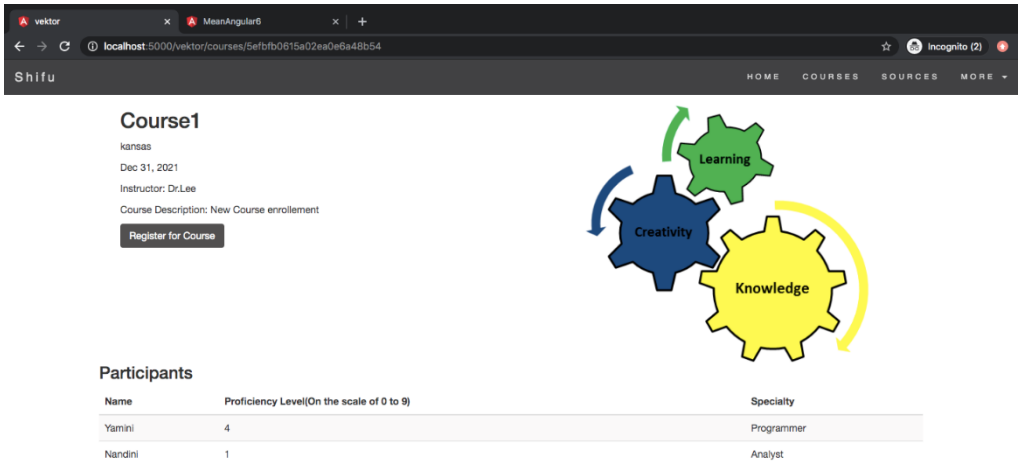
Courses Page: Faculty users can view the list of courses here. The user can view all the courses added to the platform here. Faculty can edit or delete his own course information but restricted to edit courses created by other faculties; this role based access is planned in future.

Edit & Delete of course information involves CRUD operation involved UI, service layer and Datastore.

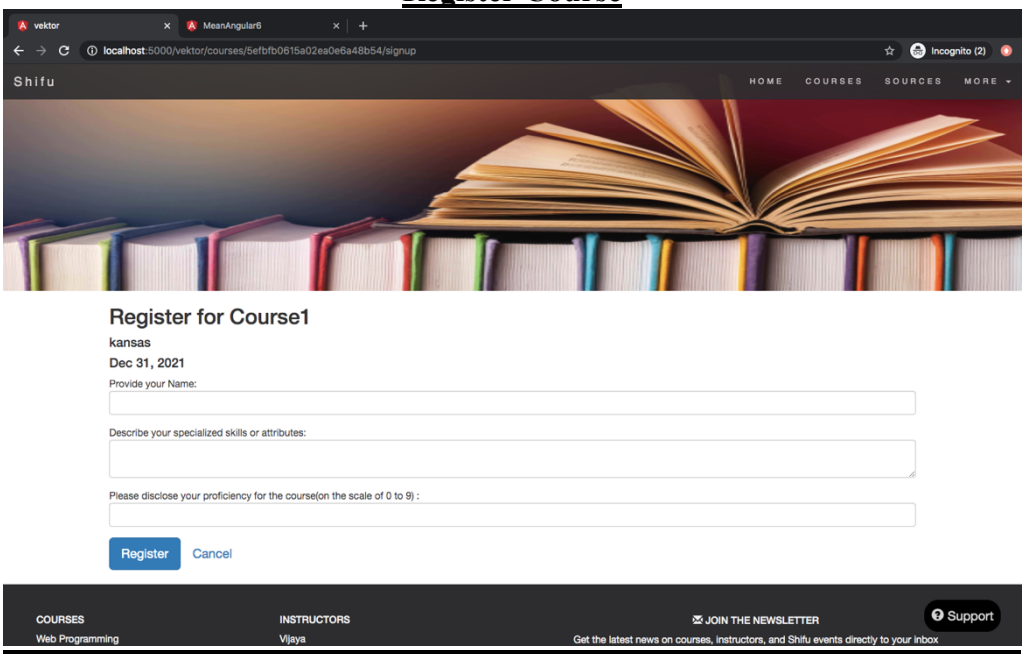


Course Registration Page: The Student users can view the list of courses and register based on the subject, schedule and location. Students can also view the list of other students (only names and skill) who has already registered for any specific course to understand the peer group proficiency level and open slots.

View Course

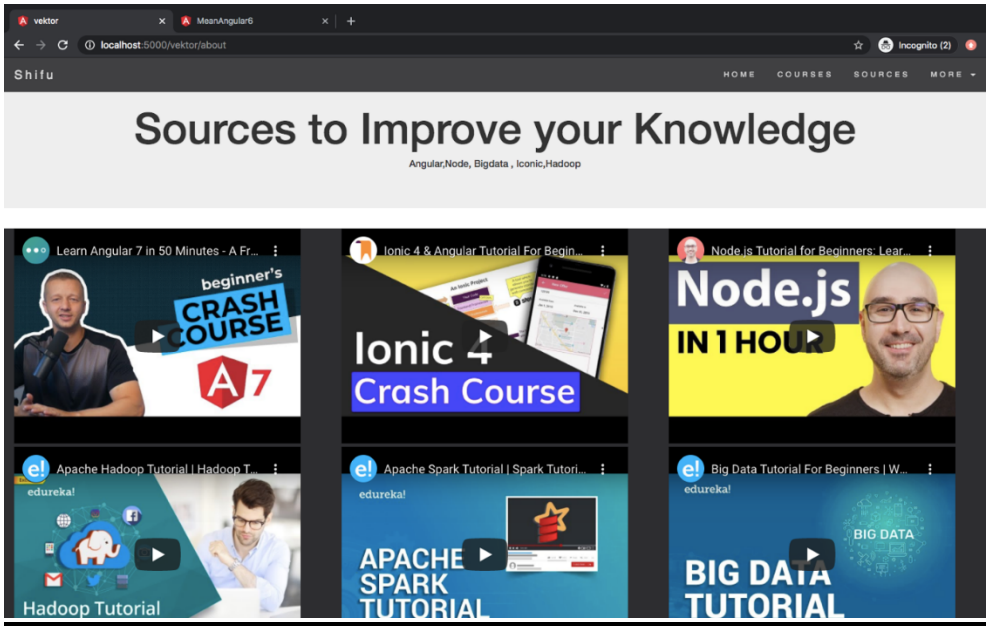


Register Course

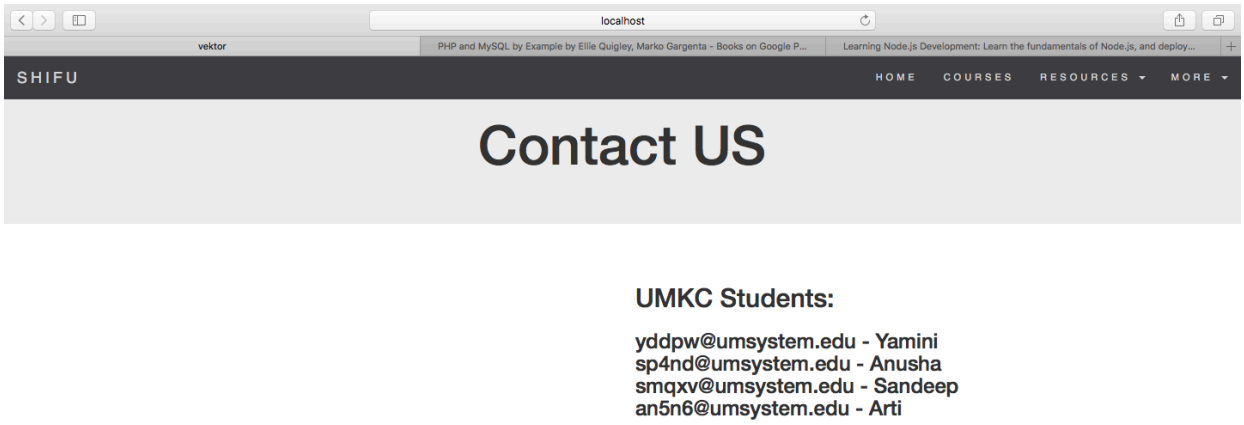


Resources Page: Students can learn from the offline and recorded videos in this page for offline education. The user can pause or perform any actions on the video.

Resources pages would be enhanced with other online resources such as books, research papers and reference links in future.

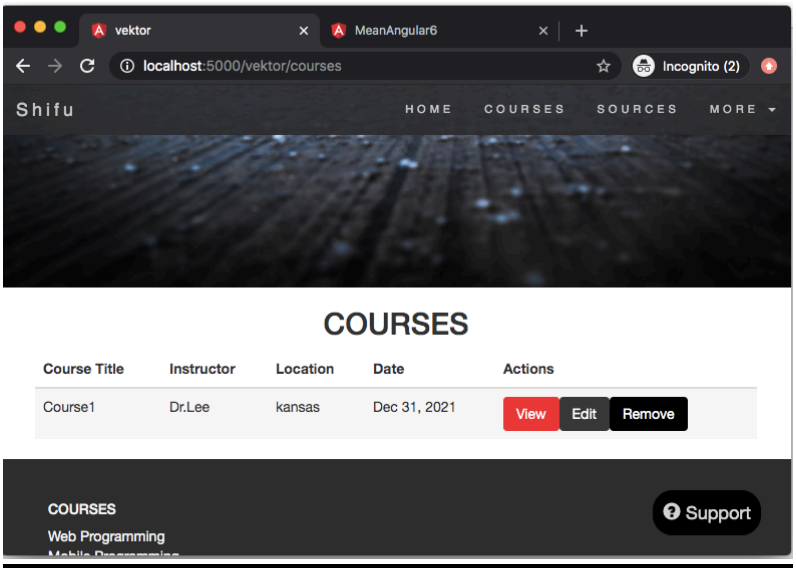


Contact us Page: Both type of users can contact both admin and academic teams through below contact coordinates.



Responsive Page: Designed to provide the Best Experience for all users. Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

Minimized Screen



Future Scope

Current implementation covers only the pilot use case that involves basic course creation, modification and enrollment with limited API integration and DB usage. This serves as the proof concept for the technical architecture and future scalability of this platform.

This platform is envisioned as community platform that would be available across the geographies as crowd sources model for education. It would serve the society in two broad opportunities,

- To improve overall basic literacy across developing countries to enable primary education through online with involvement from “Citizen Faculties”.
- Helps to deliver advanced topics to students from faculties across the globe in respective languages

Future enhancement would involve below features,

- Role based access for Students, Faculties and Admin users.
- Online Meeting features to deliver and attend lectures through virtual class room – integration with Google or Zoom.
- Feedback process for course & faculty ratings.
- Offline resources such as videos, research papers and reference links.
- Funding features for crowd funding.

Challenges

- Conceptualization of Idea and the application with focus on future business model.
- Use case design and architecture based on MEAN stack.
- Integration of APIs with Front end and DB.
- All of the team members are new to Angular and Node JS, hence there was a learning curve to build this use case.

- New to use MangoDB and MangoDB Cloud – so, faced initial hiccups while developing the application.

Learning Outcome

- Helped to understand MEAN stack well and being new to Angular and Node JS platform. With this project, learned to develop the end – to – end application successfully.
- Being new to services and REST, understood and had hands on experience about developing API based application. Also, ultimately this project experience gave us an opportunity to ideate and develop online platform focused on community learning.
- Gained enormous confidence to work on real time projects after this project and can successfully work as a team too.

Expected Project Outcome

This use case and the reference implementation is a good case study to expand the scope on both technology and functional sides. This can be developed as community online platform where scholars and students can collaborate seamlessly.

Cloud based microservices architecture will make it scalable for an enterprise level usage and can serve users across the globe.

References

- **https://developers.google.com/books/docs/v1/reference/volumes#methods**
- **https://console.developers.google.com/apis/library?project=web2020-281816**
- **https://unsplash.com/**
- **https://angular-templates.io/tutorials/about/learn-how-to-build-a-mean-stack-application**
- **https://forum.freecodecamp.org/search?q=mongodb**

Links

Application URL(Local Host) :

http://localhost:5000/vektor

GitHub Wiki Link:

https://github.com/yaminireddyd/Web-MobileProgramming---2020Summer/wiki/Web_Project(MEAN-Stack)

GitHub Project(Source Code) Link:

https://github.com/yaminireddyd/Web-MobileProgramming---2020Summer/tree/master/Web/Web_Project1/MEAN_Stack_Project

VIDEO LINK:

https://umkc.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=45d470b9-08f5-404d-8db5-abed003b04de

Acknowledgements

We express our sincere gratitude to Professor **"Vijaya Kumari"** for providing us an opportunity to do the project "Shifu" and also for the guidance and encouragement through out the project.