U-IMPACTIFY

Learn, Grow, Give
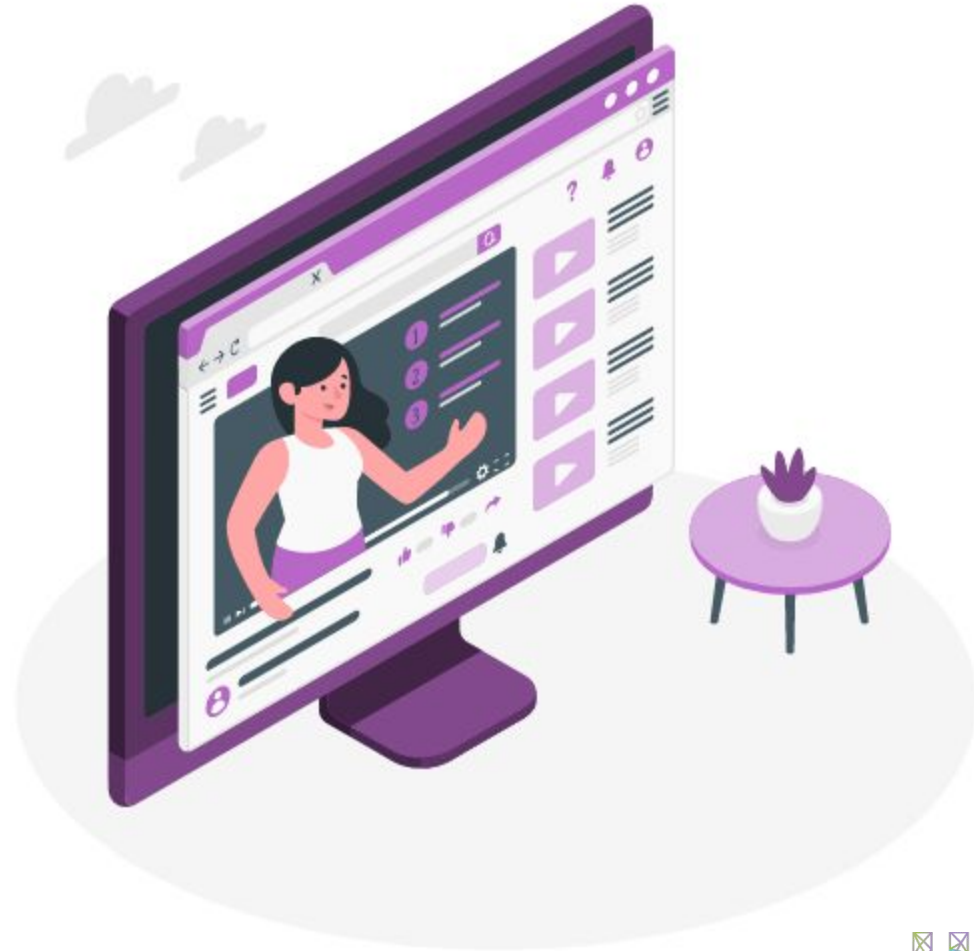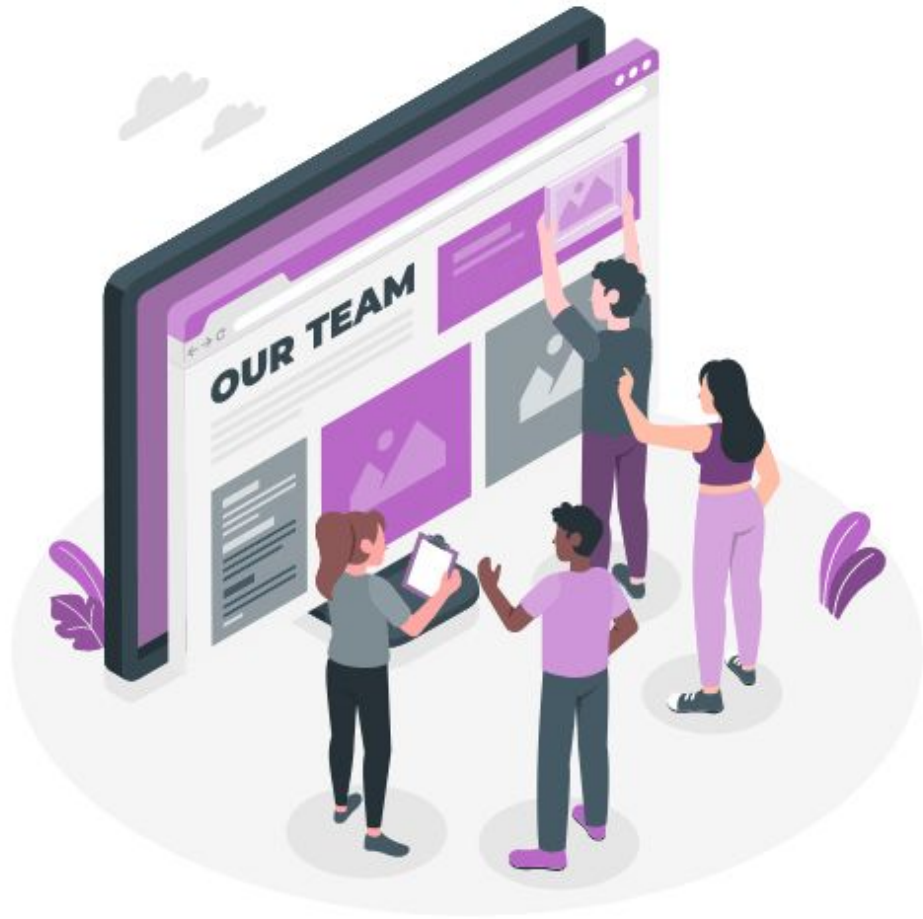
OUR TEAM

WE ARE TEAM **B**OUNDLESS

Annihilated Aryan

Big Brain zBrian

Corpse Clara

Dead Divyam

negative nancy navinn - no.

Slaughtered 5amyak

WebWork Winson
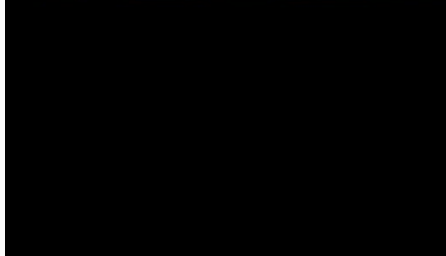
INDIVIDUAL CONTRIBUTIONS

Aryan - User Course Review, Assessments, Course Preview, Pop-Ups and lecture upload Refinements

**Brian-** Course Preview, Course Refinement, Assessments, Student Analytics

Clara - Building Backend, User and Course creation, User Profiles, File and Video Uploads, Giving Garden

Divyam - Updating Documents, Questionnaires, Giving Garden, Backend Grading, and refinements

Navinn - Sprint Meetings, Chat, Search, Course Refinement, Opportunities, Deployment

Samyak - Frontend Questionnaire, Backend and frontend Assessments, social initiative page, and refinements

Winson - Course Evaluations, Creating Course, Opportunities, Searching User/Course, Deleting Users
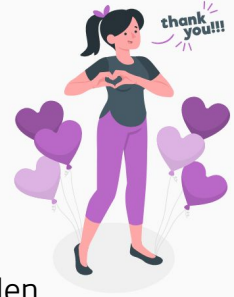
OUR PROCESS

Git Hygiene
- Branch for every sub-task
- Before a merge, a pull-request must be made and reviewed by two other members

Communication
- Discord Server
    - where all of our communication took place

Scheduling Meetings
- Daily Standups @ 10:00, used a bot if no one was available
- Someone proposes a time, and everyone confirms

Assigning Work
- Everyone gave their preferences, but decided based on individual capability and team capacity
- Flexibility amongst team members

RETROSPECTIVE

HIGHLIGHTS

- Everyone did their work, and finished their tasks on time
- Everyone was held accountable for their work
- Effective communication amongst group

DIFFICULTIES

- Learning curve, not everyone was familiar with the different technologies
- Managing other work with this project

WHAT WORKED WELL

- Live daily standups
- Group calls to discuss
- Communication frequently
- Messaging for help as soon as needed
- Incorporating a Discord Bot for standups

MAKING DECISIONS

- Discussed amongst group
- Each member had the opportunity to provide their thoughts and ideas
- Finalized when majority agreed upon decision

TECHNOLOGIES AND TECHNICAL CHALLENGES

THE TECH. STACK

Backend

mongoDB®

ExpressJS

node JS®

socket.io

THE TECH. STACK

Frontend

ANGULAR

Figma

Chart.js

ngx-file-drop

Clarity
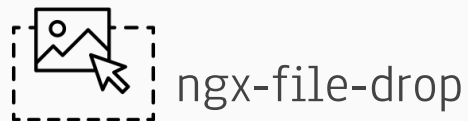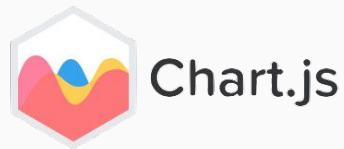
TECHNICAL CHALLENGES

CROSS ORIGIN ERRORS WITH SOCKET.IO

- Built static frontend files vs. separate web server
- Made deploying easier

OUTDATED ANGULAR WRAPPERS FOR JS LIBRARIES

- Used the core API for the library + followed documentation rather than tutorials/videos

CONNECTING BACKEND AND FRONTEND

- Ensured backend developer co-ordinated with frontend developer
- Maintained efficiency in combining backend with frontend

ENDPOINT SECURITY
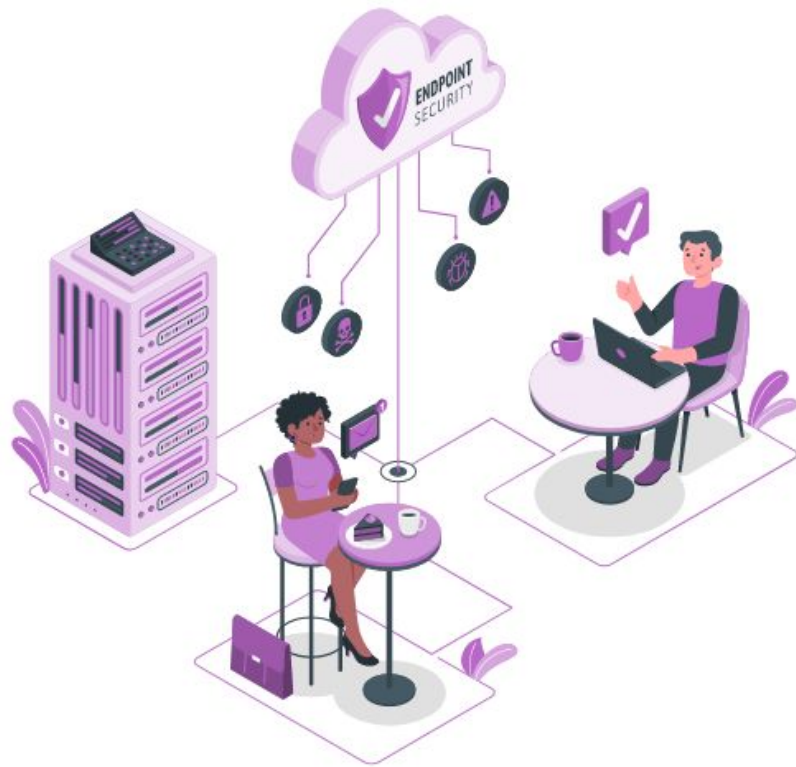
SOFTWARE ARCHITECTURE

MAIN COMPONENTS

MONGODB

Collections
- assessments
- chat
- course_uploads
- courses
- opportunities
- user_uploads
- users

FRONTEND

- Angular
- Clarity, a component library
- Chart.js
- Ngx-file-drop

ANGULAR SERVICES

Services
- User
- Course
- Chat

API ROUTES

Routes
- User
- Course
- Chat

MongoDB Database

<- GETs, POSTs, PUTs, and DELETEs data from --
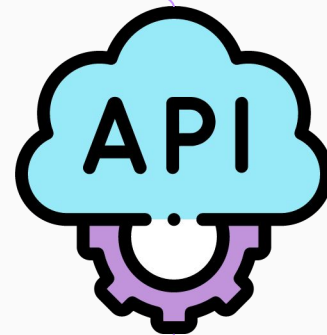
Node
Express
Socket.io

-- calls API routes ->

Angular: Services & Guards

<- sends response --

Frontend
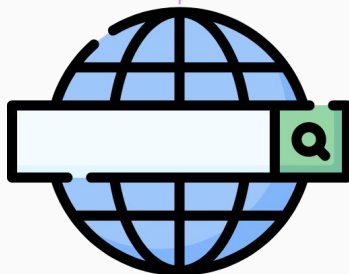
SOFTWARE TECHNIQUES

DEPENDENCY INJECTION

- Injected services into the components
- User information injected where needed

MODULAR COMPONENTS

- Encapsulated template, styles and controller logic files
- Reused throughout the core application

MVC

- Initially designed following the MVC design pattern
- Formed code following the MVC design pattern

- **Model** - Mongoose Schemas (database)
  - User, Course, Assessment, Chat, and Opportunity Models
  - Properties of data stored in each object

- **View** - HTML/CSS/TS (front-end)
  - Each component has its HTML and TS file where the TS file holds instances of models to update the HTML
  - Each component is encapsulated with its own HTML, CSS, and TS files
  - TypeScript Interfaces for models, synchronous with Mongoose Schema

- **Controller** - Angular Services/API Routes
  - Input/output function logic handled through services and routes

MODEL

UPDATES   MANIPULATES

VIEW   CONTROLLER

SEES   USES

USER

Model	View	Controller

Manipulates

Login
- header component
- email prompt
- password prompt
- login button
- side image

Signup
- header component
- email prompt
- username prompt
- password prompt
- user type prompt
- side image
- signup button
- questionnaire

User Action

UserService
- user: User
- postNewUser(newUser: any): Observable<any>
- loginUser(email: String, password: String): Observable<User|Boolean>
- getCurrentUser(): User
- setUser(user: User): void

Updates

User
- Username: String
- Password: String
- Email: String
- Type: String
- Classes Enrolled: String[]
- Classes Teaching : String[]
- Questionnaire : String[]

- comparePassword(String newPass, callBack: function): void

Interacts

UserRouter
- loginUser: POST
- newUser: POST
- updatePassword: POST
- updateEmail: POST
- deleteUser: DELETE

Updates

Mailbox
- user: User
- sender: User
- subjectLine: String

- sendEmail(message: String): void
- recieveEmail(); void

ReviewAndFeedBack
- Feedback Form
- prompt to create survey
- Discussion thread
- Manage existing section

Updates

Updates

CourseService
- course: Course
- postNewCourse(newCourse: String): Observable<any>
- postNewFile(file: any, CourseId: any): Observable<any>
- getCourseFiles(CourseId: String): Observable<any>
- getCourseAssessments(CourseId: String): Observable<any>
- postCourseAssessment(newAssessment: String): Observable<any>

User Action

Updates

Dashboard
Social Initiative
- organization profile
- navigation bar
Impact Consultant
- course
- navigation bar
Impact Learner
- course list
- navigation bar

CreateCourse / UpdateCourseContent
- Course Name
- Course Description
- Course Difficulty
- Course Tag
- File(s) Upload

Extends

Interacts

CourseRouter
- addNewCourse: POST
- getCourse: GET
- uploadDocument: POST
- getDocument: GET
- deleteFileById: POST
- getAllDocumentsForCourse: GET
- uploadAssessment: POST
- getCourseAssessment: GET
- getAllCourseAssessments: GET

View

Course
- Title: String
- Students: String[]
- Teachers: String[]
- Description: String
- Documents: String[]
- Assessments: String[]

Extends

Preview Course
- Course Name
- Course Description
- Course Difficulty
- Course Tag

Private Messaging
- Notifications
- 1-to-1 PMing

LIVE DEMO

FIN. 👋

https://uimpactify.herokuapp.com/