# CMPE 352 Spring 2020 Group 3 - Milestone 2

## 1. Executive Summary

### 1.1 Introduction

For this milestone, we have done a practice app that covers the implementing, testing, deploying an API; and providing a simple UI to interact with the API. For this task, we have learned backend and frontend frameworks and put together our collaborative work with git flow. Coding as a team has been challenging but a good learning experience for us.

### 1.2 Work done so far

Firstly, we decided on which frameworks to work with and set up our working environment. We have chosen NodeJs for backend, ReactJs for frontend, MongoDB for database. Secondly, we've created a group mail and created a MongoDB instance with MongoDB Atlas. Then initialized the backend and frontend projects.
Thirdly, we decided on which external APIs that we'll use, who will work on which API. General workflow was working with pairs with each API and writing one endpoint for each team members. These are API's that we've worked on and endpoints:

- Login / Register endpoints ( with no external API )
- Listing news endpoint ( with News API )
- Searching / Listing events endpoints ( with PredictHQ API )
- Google Scholar Article endpoints ( with Google Scholar API )

Finally, we've created a frontend to register and log in to the system, then work with the endpoints that we've implemented.

### 1.3 Road ahead

We've chosen APIs that we can use for the real project on CMPE 451, therefore we can improve the functionality of the endpoints and add new ones. Additionally, since we've implemented register and login systems, we can improve them and use in the real project

### 1.4 Challenges we met as a group

Most of us haven't worked on an API project before, therefore learning and using a new technology was a bit challenging for us. Also, due to the covid-19, we couldn't meet physically as a group and this led to a communication problem. Lastly, coding as a team with a version management system was a way of working that we weren't used to, creating pull requests, reviewing each other's code and solving conflicts was both educative and challenging for us.

# 2. List and status of deliverables

| Deliverables | Status | Due Date |
|---|---|---|
| Group Deliverable | **Done** | 27.05.2020 |
| Individual Deliverables | **Done** | 27.05.2020 |

# 3. Evaluation of the status of deliverables and its impact on project plan

### 3.1 Implementation

Implementation part was the most crucial part of this assignment. Everything for our practice application depended on it. Also note that, we lack experience as a group on API implementation. First of all, we decided on what languages and frameworks to use for our project. We decided on Node.js for backend and React for frontend implementation.

After that, we decided on which APIs and endpoints will be created. We decided on creating 4 APIs. 3 of those API use external APIs but it is not necessary to use external API for login and register. 8 people from the group worked on APIs in groups of two and 2 people worked on frontend part of the practice application using React.

Everybody created their local branches to avoid conflicts and after implementing their task, pushed their local branch to remote and merged their branch to ‘develop’ branch by opening pull request. We have successfully completed the APIs, endpoints and UI for our practice application.

### 3.2 API Documentation

API documentation can be thought of as a user manual for our APIs. We created API documentation to help visitors of our project to understand how our APIs and endpoints work. Also, after some time it is possible to forget how we write our code and understanding the functionality of our APIs can be hard by reading the code. Therefore, API documentation is also a guide and reminder for us to understand our own work.

### 3.3 Project Plan

Before this assignment and milestone we created a detailed project plan for our real project. We created a Gantt chart to determine our goals, where we are and where we aim to be at with our project. Since we have successfully completed our tasks until now, everything is going according to our project plan.

## 4. Summary of the work done by each team member

| | |
|---|---|
| Mahir Efe Kaya | Created the scholar endpoint which was essentialy created for deriving data regarding the scholar's previous articles in the google scholar using an api with the help of Budak |
| Ali Furkan Budak | Created scholar endpoint with Mahir. Prepared unit tests for scholar endpoint. |
| Ramazan Koç | I created the listNews endpoint which returns news about given user id. I prepared milestone 2 group report. I reviewed Ali furkan budak's scholar endpoint pull request. |
| Yusuf Bayam | Established connection to MongoDB from our application with Barış. Implemented Registration and Login APIs with Barış. Implemented granting token to user after successful login. Implemented two unit tests for Login functionality. I contributed to group milestone 2 report with Ramazan and Yahya Bedirhan. |
| Furkan Cansever | I implemented Events API filter endpoint. I prepared the unit test to test filter endpoint and Geo Location API It is used for filtering events using two parameters (radius,place). I used two API that are PredictHQ and Geo Location API. |
| Yahya Bedirhan Pak | I've initialized the backend project and created the project structure.<br>I've initialized the frontend project with create-react-app.<br>I've created the MongoDB database instance from MongoDB Atlas.<br>I've written getRelatedKeywords, getConferences methods, which use two APIs: Datamuse and PredictHQ.<br>I've written events/search endpoint with using those methods.<br>I've written tests for those methods and events/search endpoint.<br>I've contributed to writing group milestone 2 report.<br>I’ve dockerized and deployed both frontend and backend. |
| Hande Karabul | I fixed Login page.Register, Events endpoints are reviewed. |
| Mustafa Küçük | Registration, Login, Paperapi and Home pages are implemented.<br>Login, Register and Events endpoints are reviewed.<br>Using http client, frontends for endpoints are implemented. |
| Fatih Akgöz | I created listNews endpoint and the unit tests for this api with Ramazan. |
| Barış Başmak | Established connection to MongoDB from our application with Yusuf. Implemented Registration and Login APIs with Barış. Implemented unit tests for Registration functionality. |

# 5. Evaluation of tools and processes

Firstly, for the communication tools, we've used Discord for regular meetings. It was adequate for discussing future works and distributing the tasks. Also we used WhatsApp for fast communication and generally moved the conversation to Discord for further discussion. Secondly, we used git for version management system. We've tried to work according to git flow, everyone opened a separate branch and merged it into **develop** branch with pull requests. This cycle has been managed well between team members, we've tried that each pull request is reviewed at least one member.
Lastly, for the frameworks that we used, most of us wasn't experienced with any of the popular frameworks such as Django, NodeJs, JavaSpring. Since it was said to be easy to learn, we decided to start with NodeJs and followed some tutorials. Since Yahya has some experience with creating a NodeJs project, he has initialized the project structure and we put our work on top of it.  For the database, we aren't used to work with relational databases, therefore we decided to work with a NoSQL database, therefore we've chosen the most popular one, MongoDB. It was easy to work with and adequate for this practice app. For the frontend part, we decided to work with a popular framework to have a general knowledge and get familiarized with fundamentals of frontend technologies, therefore we've chosen ReactJs.

# 6. API documentation and URL

**Frontend URL:** http://13.58.21.212:3001/
**Backend URL:** http://13.58.21.212:5001/

- POST - Login User

  User sends email and password to the system as a request to the server.
  - Endpoint : /api/login
  - Request body : JSON
  - Required parameters
    - Email : string
    - Password : string
  - Possible error codes :
    - 400 : Bad request
    - 401 : Invalid email or password
  - Sample Request Body :

```
{
    "email" : "celalbaba33@hotmail.com",
    "password" : "123456"
}
```


- Sample Response :


```
{
    "message": "Login Success!",
    "user_token":
        [redacted]
}
```


- POST - Scholar
  - Endpoint: /api/scholars/:
  - Request body : JSON
  - Required parameters
    - name: String
    - bio: String
    - area: String
    - scholar_id: String
    - articles: List<Article>

      Article : {

      “contributors”: String,

      “url”: String,

      “name”: String

      }

- GET(all)- Scholar
  - Endpoint: /api/scholars
  - Request body : No need for parameters
  - Sample Response:


- GET- Scholar
  - Endpoint: /api/scholars/:id
  - Request body : JSON
  - Required parameters
    - _id : String

- PATCH - Scholar
  - Endpoint: /api/scholars/:id
  - Request body : JSON
  - Required parameters
    - _id : String
    - bio: String
    - area: String
    - scholar_id: String
    - articles: List<Article>

      Article : {

      “contributors”: String,

      “url”: String,

      “name”: String

      }

- DELETE - Scholar
  - Endpoint: /api/scholars/:id
  - Request body : JSON
  - Required parameters
    - _id : String

- POST - Register User

User sends email and password to the system as a request to the server.

- Endpoint : /api/register
- Request body : JSON
- Required parameters
  - Email : string
  - Password : string
  - Name: string
- Possible error codes :
  - 400 : Bad request
- Sample Request Body :

```
{
    "password": "12345678",
    "email" : "NeZpPK33@yohey.com.uk2",
    "name" : "osmanaga"
}
```

- Sample Response:

```
Success!
```

## • Getting News

https://github.com/bounswe/bounswe2020group3/blob/develop/practice-app/API_DOCUMENTATION.md#getting-news

## • Filtering Events

https://github.com/bounswe/bounswe2020group3/blob/develop/practice-app/API_DOCUMENTATION.md#filtering-events

## • Searching Events

https://github.com/bounswe/bounswe2020group3/blob/develop/practice-app/API_DOCUMENTATION.md#searching-events