



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## Milo - A College Connect Platform

<sup>1</sup>Jaydev Sharma, <sup>2</sup>Mruthyunjay Gurav, <sup>3</sup>Himanshu Singh, <sup>4</sup>Shaina Gupta

<sup>1,2,3</sup>Department of Computer Science Engineering, Lovely Professional University  
Phagwara, Punjab, India

<sup>4</sup>Assistant Professor, Department of Computer Science Engineering, Lovely Professional University  
Phagwara, Punjab, India

**Abstract:** Milo is an all-in-one system that helps college students to make the most out of their online college life. Milo takes out and combines the good parts of the existing system like Q and A forums Quora Discussion forums Reddit and Social networks and optimizes it to cater to the needs of the students. This would help the student to stay connected and get the answers the student needs directly from their Faculty members or their peers, talk and discuss easily, stay up to date about events and jobs. These papers will go over the flaws in the existing solutions when talking from the perspective of a student, the result of the survey that was conducted, Describes our solution in detail, technical specifications, and conclusion

**Keywords – Graph QL, MongoDB, MAGIC, PASSWORDLESS, Microservices, Next Js**

### I. INTRODUCTION

Milo is a platform that facilitates and ensures to bring up better engagement between the students, facilities, and the world that students have to be prepared for. It's the platform that is meant to narrow the gap between what is expected and the reality by connecting the students directly with the faculties and industry experts.

Milo is currently a web-based platform with features for staging students to openly discuss, enthusiastically ask questions, express their thoughts, build engaging communities, and organize fun, informational events and off-course meetings and interacting with industry experts. However, as much as it is to encourage learners for better engagement it is also important to ensure that there is no fear of expressing, approaching or the question of struggling through "how" and "what if" pinching the young curious minds. So, Milo not only combines the best part of all the existing platforms but works on maintaining the identity of the students while staying well connected and engaging frictionlessly.

Raising awareness about the expectation, requirements of the industries and being prepared for them has always been a challenge. Milo narrows that gap by allowing students to directly interact with the people from the industries. Apart from all these, Milo believes in community-based learning. So, it also aims at encouraging enthusiasts for organizing events, career fairs and also make it easy for them to find jobs/internships.

### II. PROBLEM STATEMENT

**Need for Unified Socialism among students:** The rapid increase in the emergence of web-based technologies has resulted in exponential consumption and sharing of data across the world (Around 59.5% of the world population uses/has internet today, in 1995 it was less than 1% and these factors are increasing every second). Among these technologies rapid and vast adoption of social media, technologies are changing how we exchange information, Social media sites (SMS) are a virtual platform that helps people to make new connections, exchange information and improve on friendly relationships which are necessary for students, learners, and enthusiasts to meet new like-minded people. One such example is Facebook, being the largest social media platform has approx. of 2.7 billion active users at a given time.

These numbers are huge -there are around 7.8 billion people in the world with 4.8 billion of us online Likewise, other social media platforms including YouTube, Instagram, Twitter also have more than one billion users each. This means social media platforms are used by one-in-two people in the world. Among these 90% of the users are from the 18-26 years age group. Many studies (Barnes & Laird, 2012; Carroll & Kirkpatrick, 2011; Gok, 2015; Nehls & Smith, 2014; O'Keeffe & Pearson, 2011) were conducted to determine the positive and negative impact of social media sites. Salas & Alexander (2008) stated that social media sites provided students with a platform to discuss and share academic issues, class materials, and ideas Schill (2011) reported that social media sites cause many severe negative impacts (behavioural changes, anxiety, psychological effect, mental health problems, severe loss of personal productivity, sense of guilt and crisis, stress, etc.) on adolescents. Gok, T. (2016). Published in International Journal of Research in Education and Science (IJRES), 2(1), 85- 93. conducted research that examined the effects of social networking sites on students' studying and habits. The research concluded with the findings that many students have an interest in social networking sites.

This interest leads them to get addicted. The findings also stated that Social networking sites negatively influence students' habits, grades, and socialization. With all these findings we come to know that there is a necessity for an online forum that rather aims at unifying students, learners, and enthusiasts, a unified social platform that not only empowers the underrepresented but also provides a way to open and community-based learning. While there a lot of existing solutions do serve their purpose, but their purpose is different from what we are targeting, and/or it covers a broad spectrum.

### III. ISSUES WITH EXISTING SOLUTION

**A. Chegg Platform:** A reputed brand to get your doubts cleared from the experts in that particular field, but at a higher cost and not everyone can afford to get a paid subscription.

**B. Quora Platform:** A free-to-use one of the most popular Q and A platforms covers a broad spectrum of topics from arts to science to movies and games. Because of this huge spectrum, the students might not get the answers they are looking for Also in recent times, a huge amount of questions, and no dedicated person to answer have left many questions unattended.

**C. Facebook Platform:** Facebook pioneered and revolutionized the way people connect online. Facebook is used widely by students in completing their basic needs and interaction. Soon, Its frequent use brought an addiction in students and deviated from their basic goals in life. (Revista De Administratie Publica Si Politici Sociale 3 (7), 95, 2011) Unavailability of a niche full-stack Platform: There isn't any full-stack platform that incorporates all these platforms and narrows the focus down to students, their academic, and overall development. A student has to sign up for different platforms and pay for different services. Continuously managing different platforms with their broad spectrum is quite distracting and less fruitful.

### IV SURVEY

What's your current status ?  
13 responses

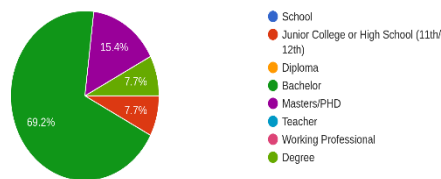


FIGURE 1: Current Status of students

**Inference:** The Demographic is strongly towards the students of bachelors accounting to 68.2% and then followed by masters PHD and Junior College and Degree students of bachelors accounting to 68.2% and then followed by masters PHD and Junior College and Degree

Do you feel comfortable in asking questions to your faculty in front of the class(Online and Offline)  
13 responses

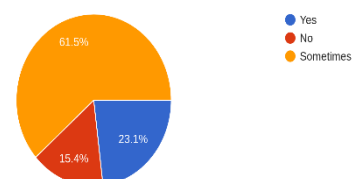


FIGURE 2: Comfortable in asking a question?

**Inference:** Around 15.4% of the students find it difficult, the rest 51.5 % hesitate but can ask and 23% of people feel completely comfortable asking questions

How easy is to get your Doubts cleared during Covid  
13 responses

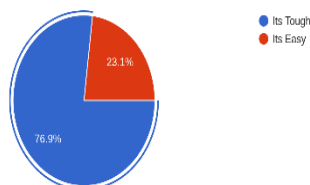


Fig. 1.FIGURE 3: Current Status of students

**Inference:** The majority of the population says it's tough to get doubt cleared during the Covid

Do you think such a platform would be of use where Students/Faculties can ask doubts, discuss and have a social network of their own College/University  
13 responses

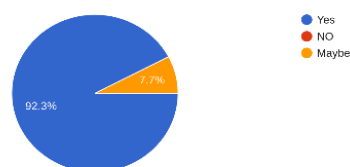


FIGURE 4: How platform useful is?

**Inference:** Around 92% of People find our solution useful to have a platform where the Students and Faculties can ask doubts, discuss and have a social network of their own.

## V. OUR SOLUTION

Milo (College Connect) is an all-in-one system that takes out and combines the good parts of the existing system like Q and A forums (Quora), Discussion forums (Reddit), and Social networks and optimizes it to cater to the needs of the students. This would help the student to stay connected and get the answers the student needs directly from their Faculty members or their peers, talk and discuss easily, stay up to date about events and jobs. Students generally struggle to get the answers or get their doubts cleared on Q and A platforms or they take so much time that the need for that has already passed. Most of the time the subject matter expert or faculty are better equipped to clear the doubt.

The Platform aims to bridge the gap between the students' doubt and the field's faculty or expert in the college. This also comes with the added advantage of faster doubt clearing time and a place for other students to search and clear the most common doubts. Students find it hard to get a platform where they can talk with the masses and discuss things that matter. A discussion could be about an upcoming cultural event, a recruitment drive in some club, or discussing a college project. The platform aims to build a live discussion platform where the college can openly discuss like an online forum with the added feature of staying anonymous, the Introverts who are scared to ask could also ask freely. This could bring upon a sense of confidence and help them open up. (European Journal of Business and Social Sciences 6 (11), 01-18, 2018)

Milo also incorporates a direct messaging service that connects each member of the college/university with others and also makes it easy to search for a person to talk to.

Apart from all the Academics, Milo also aims at managing and informing the students about the upcoming events in their college/area that they could attend and a Job portal that helps them find relevant jobs/Internships/gigs.

## VI. TECHNOLOGY USED

**A. Graph QL:** Graph QL is a query language for API developed by Facebook's alternative to REST for communication between browser and a server (open-sourced in 2015) designed to keep in mind to build robust client applications by providing an intuitive, flexible syntax and system for describing their data requirements and interactions, it also includes a runtime to fulfill those queries as well. It is designed to reduce the number of API calls to the server and return only the data that is requested, thus giving clients the power to ask for exactly what they need. Graph QL also helps in documenting the API while providing a complete and understandable description of the data in our API.

The GraphQL philosophy is quite different from that of REST (Representational State Transfer). REST is resource-based.

Every resource, for example, a "blog" has its address that identifies it, for example, /blogs/12. All operations done to the resource are done with HTTP requests to its URL. The action depends on the used HTTP method.

This resource-based pattern of REST works well in most situations, however, it results in difficulties at times in many situations.

For example, our application needs a list of all the blogs that were written by the user and all the comments to the respective Blog. In such a situation if the server implements REST API, then we must make multiple HTTP requests from the browser before we have all the data we wanted (probably the endpoints would be /users/:id, /blogs, /comments).

Not to forget that the request would return a lot of unnecessary data, and surely the code to return this data would be complicated.

Result of query:

```

query FetchUserBlogsQuery
{
  user(id:132456)
  {
    username
    email
    blogs
    {
      title
      description
      comments
      {
        comment
        writtenBy
        {
          id
          username
        }
      }
    }
  }
}

```

FIGURE 5: Example Graph QL dataset snippet

```

{
  user: {
    username: "John Doe",
    email: "johndoe@mail.com",
    blogs: {
      title: "Blog title",
      description: "Blog description",
      comments: {
        comment: "Comment on blog",
        writtenBy: {
          id: 13213,
          username: "Jane Doe"
        }
      }
    }
  }
}

```

FIGURE 6: User Data Snippet Example

**B. MongoDB:** MongoDB is a NoSQL document-based, general-purpose, distributed database built for high performance and scalability. As it is document-based, it stores data in JSON-like documents as it's a more powerful and most natural way to think about data, it is much more expressive than the traditional row/column model in a relational database.

NoSQL (not only SQL) are non-tabular, storing and relating data differently than relational tables. NoSQL databases come in many types based on their data model, a few of the main types are document, key-value, graph, and wide-column.

They provide a flexible schema and are designed to scale easily. MongoDB being a document-based database provides all the power of relational databases and more capabilities like distributed multi-document ACID transactions with snapshot isolations, has support for joining queries, aggregating data between collections.

MongoDB stores data as JSON (JavaScript object notation) documents in a binary representation which is called BSON (Binary JSON). Unlike most other databases that store JSON data as primitive strings and numbers, BSON encoding extends the representation to include additional types like date, floating-point, int, long, decimal128, etc... This makes it much easier for representation to include additional types like date, floating-point, int, long, decimal128, etc... This makes it much easier for applications to store, sort, and compare data. With its document data model, there is no need to decompose data across tables, run expensive JOINS, or integrate a separate ORM layer. Data that is accessed together are stored together, so one must write less code.

Some of the other features of MongoDB are:

- Code-native access
- Flexible Schema
- Change friendly design.
- Powerful querying and analytics.
- Easy horizontal scaling
- Multi-cloud, global database
- Distributed Architecture: Scalable, Resilient and Mission Critical

**C. Next Js:** Next.js is an open-source, zero-config React front-end development web framework that enables functionality such as server-side rendering and generating static websites for React-based web applications. Next.js has an opinionated way of rendering content on the client. The views can be client-side rendered, server-side rendered, statically generated during the build process. Next.js also provides different ways to query/fetching data depending on where the data is required and rendered. Apart from all these features out of the box Next.js also provides various other features like.

- Incremental Static Regeneration
- Automatic Static optimization
- Built-in CSS support
- Image optimization
- Static file serving
- Fast-refresh
- Support for typescript
- Internationalization
- File-system routing
- AMP support

**C. Password less authentication with Magic:** With trillions of people dependent on a wide range of technologies and applications. Keeping the data secured, authenticating and authorizing data access has always been a huge challenge for developers and as these systems depended on unique identifiers, passwords users were faced with other sorts of challenges to remember all those user credentials. Overwhelmed by the sprawl many users take risky shortcuts like using the same passwords/credentials between all the platforms or saving those credentials onto the platform itself. These methods that require passwords are inherently vulnerable by their very nature. Poor passwords account for 81% of all security breaches since over 59% of people re-use their passwords everywhere [1]. These cost companies a good fortune of up to \$240,000 for every 1,000 records compromised [2] -increasing risk and liability for a company.

Worldwide spending on cybersecurity is projected to reach \$133.7 billion in 2022 [3], and the average cost of a breach has skyrocketed to around \$3.92 million as of 2019 [4]. One such example is Equifax which was breached in 2017, the company is still paying off the \$4 billion damage in total.

Magic addresses many of these underlying security challenges in the world of identity and packages them in an extremely accessible product for both developers and users. It provides a key-based identity solution built on top of the Decentralized Identity (DID) standard. Where users' identities are self-sovereign by leveraging blockchain public-private key pairs. These key pairs are used to generate zero-knowledge proofs to authenticate users instead of having to rely on users providing passwords to Magic or any identity provider [3]. By leveraging elliptic curve cryptography, these proofs can be used to securely communicate with developers' own resource servers. Since Magic's authentication protocol is based on key pairs provided by decentralized blockchain networks, it is platform-independent and thus able to provide authentication service without having to rely on centralized identity providers [5].



Magic makes use of Hardware Security Modules (HSMs) provided by Amazon Web Services' Key Management Service (AWS KMS). User dedicated master keys are stored on the HSMs, which are generated using AES-256 with 384-bits of entropy. The master keys never ever leave the hardware as they are meant to be locked inside and are unable to be exported, all encryption and decryption operations happen inside the hardware itself. User's private keys are encrypted by these hardware-based user master keys, which means that attackers need to gain access to this hardware to be able to retrieve the keys and are forced to stay within magic's adversarial infrastructure -which is indeed capable to detect and monitor attackers progress [6].

**E. Apollo federation:** Federation set of tools to compose Schemas into a single data graph. when we use apollo federation we have multiple implementing services and each of those services is going to have their graphql Schema and front of those services gateway that composes these schemas into a single federated data graph so that are our web client the web clients then query a single graphql API without needing to worry about the implementation details of the services. So the reasons we use apollo federation it is one of to create a graphql API it allows us to expose one data graph, potentially with multiple data source s behind it ,like a rest API and a SQL and we can query the data that reflects the relationship between the nodes in the graph.

However, as we expose more and more objects and their relationships through various types and fields it doesn't take long for even a moderately complex app to require a large number of type definitions on turn it becomes difficult For multiple people or teams to collaborate on building the API and and graphql api becomes a single vulnerable point of failure in a system to a solution than be a create a distributed graphql architecture break up the schema into multiple smaller services for development purposes but then recombine each separately managed portion of the schema into a gateway api so that client application can remain agnostic to these divisions to querying data before apollo federation there was an existing solution to do this called schema stitching. we are talking about a few key federation concepts the first is the notion of

**A. Entities:** The entity is a type that you define canonically in one implementing service and can then reference and extend in other implementing services entities are a core building block of a federated graph[6].so the things that allowed us to split schemas based on separation of concern rather than by types alone and we create entities using an add a key directive in Our schema.

**B. Referencing:** Other services that reference the entity will only able to return a representation of that entity from any corresponding resolver so the gateway is going to have to hand this representation of to the entities originating service which means we need to provide which means we need to provide what's known as reference resolver in that originating service to fetch the for the specific entity.

**C. Extended:** We can add even additional fields to a type originally defined in another service by extending that entity in a referencing service.

#### **Apollo Server Libraries:-**

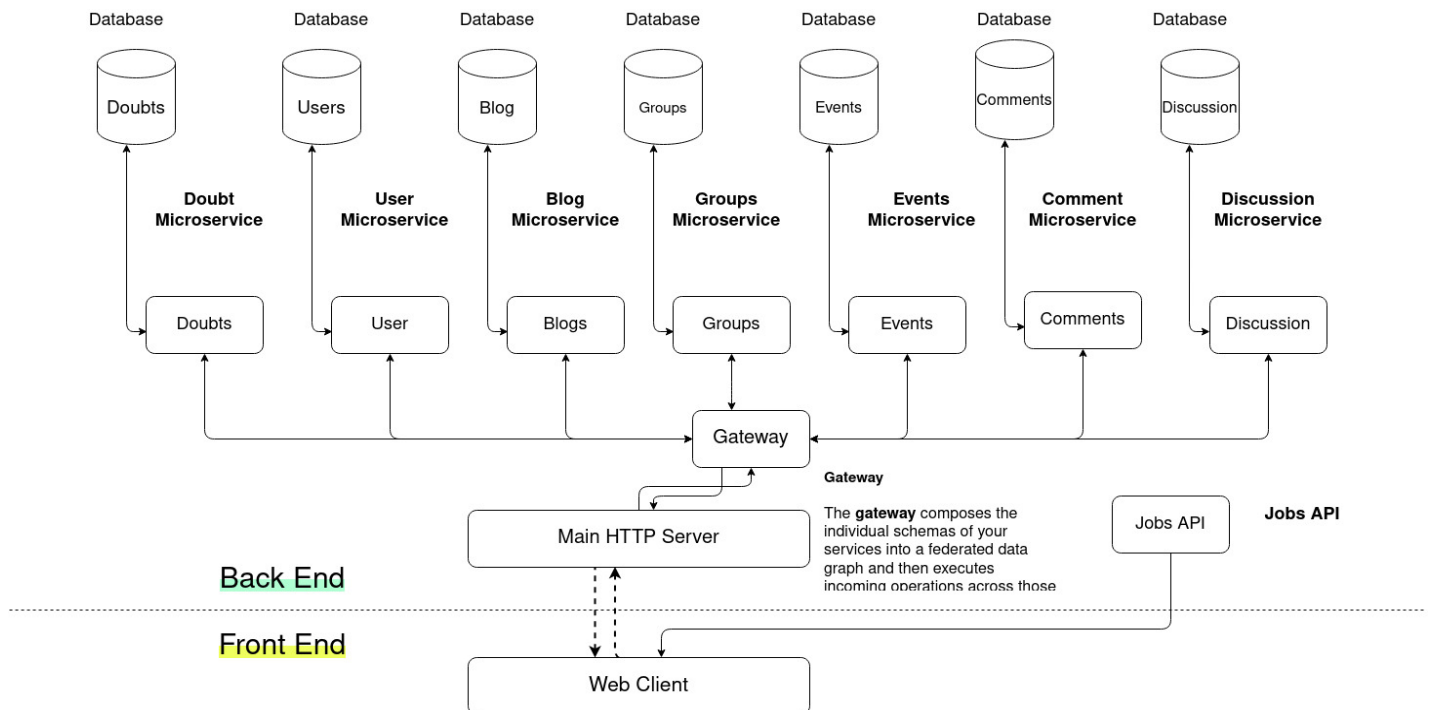
Apollo/federation: Apollo federation allows us to make our services schemas federation ready.

Apollo Gateway library: Apollo gateway library is going to help distribute incoming graphql API requests to the underlying services

Apollo server: Apollo server need an instance of apollo server for the gateway API and each of the services we create.

**F.API Gateway:** Application programming interfaces or api is a common software interface that allows two applications to communicate most modern apps are built using api as api proliferate additional infrastructure is needed to secure scale and accelerate api traffic enter the api gateway which is the single point of entry into modern api enabled application.

These can be microservices-based apps today's modern enterprises rely on API to provide more compelling customer services communicate with partners and suppliers and connect internal applications it's not uncommon to many companies for api usage making it critical source of revenue the api gateway job is to protect this critical api traffic this improve developer productivity and reduce the risk of downtime or slow performance like a load balancer.



**FIGURE 7:** Software Architecture

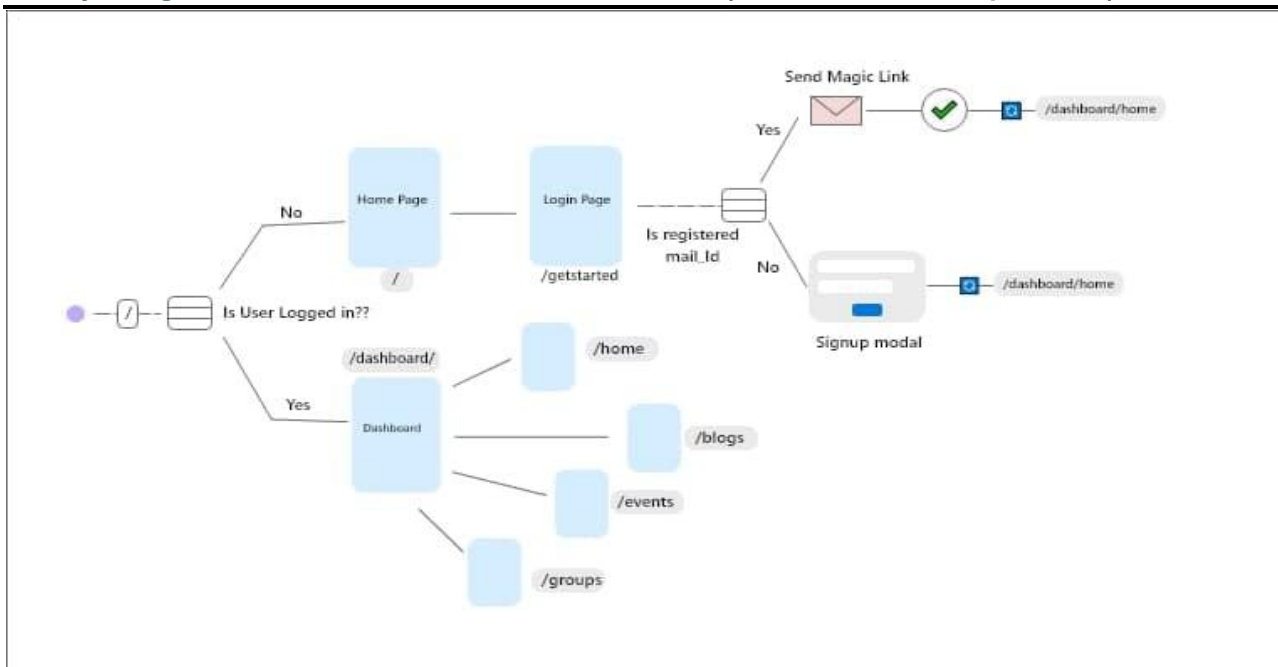
#### A. Components:

1. **Microservices:** Each Microservice (Doubts, Users, Blogs, Groups, Events, Comments, Discussion) consisting of a model, its database, and its own controller. These microservices provide data to the Gateway. These microservices are independent of one another and failing of one microservice won't take down the whole platform.
2. **Gateway:** The gateway composes the individual schemas of your services into a federated data graph and then executes incoming operations across those. A single operation requires calling multiple services. or call multiple microservices.
3. **Database:** Database is nothing but collection of data and information that is organized in such a way that it can be easily accessed managed and updated and DBMS is a software that handles the storage retrieval and update of data in a computer system some popular DBMS are MYSQL SQL Server Oracle etc. file processing system has disadvantages like data redundancy data inconsistency insurable data unstandardized data insecure data etc. DBMS store all data at one place and all the application programs that require that data can refer to the same Central place this centralization makes sure that data remains consistent and secure data redundancy leads to data inconsistency multiple mismatching copies of same data is known as data inconsistency.
4. **Main HTTP Server:** Gateway sits on this main HTTP Server which handles the incoming requests from the client-side.
5. **Web Client:** This is any Web Client such as Chrome, Firefox, etc.

#### B. The Flow:

**A. User Logged in:** If a user logged in then it will move forward to a dashboard page. User can see all the blogs created by other users. as well as the user has four choices. it will move on a Dashboard. user can see profile blogs and many more. also user will get four choices to move out of a home page it contains other user blogs or blogs page it shows other user blogs and as well as user can also create a blog using this page. or events page user can see events are going on. or a groups page groups are shows user are add-in.

**B. User Not Logged in:** If a user is not logged it will move to the home page. user can see other user blogs. but not create her blogs see other user announs so if a user wants to create her blogs and follow other user and put a comment on other users blogs first user Signup. then go to the Login page after login page sends a magic link to the user mail id and verify user through mail id it. then user go to the dashboard. and create her blogs and use all functionality of Milo app.



**FIGURE 8.**Flow of App

## VII CONCLUSION

Youths being the most important and dynamic segment of a segment of brighter better future.it is as much equally important that they spend their time wisely. There are a lot of and some students are introverts. they are afraid of teachers to ask questions. so this platform gives them a opportunity to ask the doubts annoums and also students approaches many teachers and alumni for the future goals and her doubts.

## VIII REFERENCES

- [1] Gok, T. (2016). The effects of social networking sites on students' studying and habits. International Journal of Research in Education and Science (IJRES), 2(1), 85- 93.
- [2] A MongoDB White Paper, MongoDB Architecture Guide: Overview October 2020
- [3] A Magic White paper: A Key-Based Authentication System For Self-Sovereign Identity Fortmatic Inc. March 23, 2020
- [4] Performance Analysis of Graph QL and RESTFUL in SIM LP2M of the Hasanuddin University
- [5] Research on Integrated Authentication Using Password less Authentication Method, 2017 IEEE 41st Annual Computer Software and Applications Conference
- [6] Apollo Federation Docs, Documentation Home, Explore the Apollo platform, <https://www.apollographql.com/docs/>
- [7] Next js Documentation, nextjs.org/docs