STUDYSPACE

ALI HASSAN
TAREK MOHAMED
OMAR KHALED
OMAR NOURELDEIN
ZEYAD SALEH

Table of Contents

Executive Summary:

Student Development Team:
Ali Hassan
Omar Khaled
Omar Noureldin
Tarek Mohamed
Zeyad Saleh

Background

StudySpace seeks to offer a better and more optimal solution for E-learning, Group studying and general student to student and student to instructor communications. StudySpace creates an online studying community that is accessible by both iOS and Android. it allows users to work together to achieve a higher than ever learning curve by deploying a social media like structure of communications alongside its core e-learning functions.

Project Vision

Our team's project vision is to use to use the newest mobile application development technologies and tools to build a fast, efficient and reliable mobile application that allows submission of courses, creation of studying groups and full communications between all levels of users.

Project Deliverables

Our application is to be published publicly on both Apple App Store and Google Play Store to be available on both iOS and Android mobile devices.
Student Development Team

Ali Hassan serves as a project manager and a front-end developer. He is a fourth-year student majoring in information Systems. He is seeking a career in Flutter IO Development.

Omar Khaled is a senior Business Information Technology student majoring in Information Systems. He serves as a data controller in the application who were dealing with the back-end developer to help send and receive data from the server.

Omar Noureldin serves as a back-end developer handling database, data storage and application/database online webhosting communications. He is a senior Business Information Technology student majoring in Information Systems. He is seeking a career in back-end development.

Tarek Mohamed serves as the documentation lead. he is a fourth- year student majoring in information Systems. he will continue pulsing his interests in Ul/UX design after graduation.

Zeyad Saleh serves as a graphics designer and UI/UX designer. he is a fourth-year student majoring in Information Systems. He is seeking a career in graphics design.

problem definition:

We lived most of our life as students ,and we still learning every day. If you are student at primary school, high school or postgraduate student or even a self-learning student you are searching for knowledge and seeking for help. Being in a community where you can share your ideas ,find solutions and learn new things is important. Technology had become more advanced than ever. Almost everyone today is using smartphones and tablets. Technology could help in delivering what people need to learn and what could help make a better future for everyone.

objective of the course:

a. Make all Student together
b. Make It Easy to Learn.
c. Make education easier.
d. Study at your own base.

used tools:

1- Flutter

Flutter is Google's portable UI toolkit for building beautiful, natively-compiled applications for mobile, web, and desktop from a single codebase (Dart language).

Fast Development

Paint your app to life in milliseconds with stateful Hot Reload. Use a rich set of fully-customizable widgets to build native interfaces in minutes.

Expressive and Flexible UI

Quickly ship features with a focus on native end-user experiences. Layered architecture allows for full customization, which results in incredibly fast rendering and expressive and flexible designs.

Native Performance

Flutter's widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts to provide full native performance on both iOS and Android.

Who's using Flutter?

Organizations around the world are building apps with Flutter.

See what's being created

Alibaba.com

Tencent 腾讯

Google

Google Ads

2- Android Studio:

Android Studio provides the fastest tools for building apps on every type of Android device.

3- Visual Studio Code 1.34:

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

4- GitKraken:

GitKraken is a Git GUI client for Windows, Mac and Linux. It helps developers become more productive and efficient with Git. It's free for non-commercial use.

5- AndroidSDk & FlutterSDk:

A software development kit is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

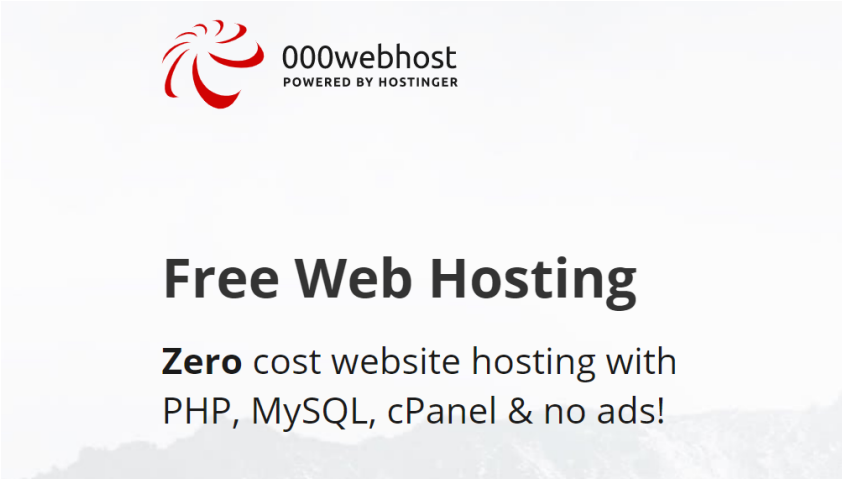
6- GitHub 2.21.0:

GitHub, a subsidiary of Microsoft, is an American web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features.

7- RealtimeBoard(Miro):

RealtimeBoard is a simple and user-friendly collaboration and whiteboarding solution for project leaders, designers, marketers, developers and creatives. ... Last, but not least, RealtimeBoard is an all-in-one collaborative platform that helps teams across the globe create the next big things.

8- 000webhost:



000webhost offers web hosting with cPanel, PHP & MySQL. StudySpace's database is hosted on 000webhost which is used to provide a live online MySQL relational database via phpMyAdmin along with the probability to create PHP files acting as API's which connects the hosted database with StudySpace.

9- Oracle's MySQL Workbench 8.0:

Used by the backend developer for offline testing, Importing/Exporting to & from 000webhosting's phpMyAdmin and for viewing the database's ER Diagram

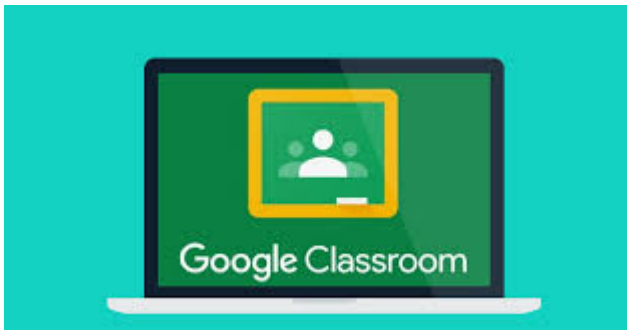
10- phpMyAdmin:

phpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services

11- Firebase:

In short Firebase is platform which allow to build web and mobile applications without server-side programming language. You can store user's data on its real-time database which sync data among user's data in no time.

Similar Apps:



Google Classroom is a free web service developed by Google for schools that aim to simplify creating, distributing and grading assignments in a paperless way. The primary purpose of Google Classroom is to streamline the process of sharing files between teachers and students.

Google Classroom ties Google Drive, Google Docs, Sheets and Slides, and Gmail together to help educational institutions go to a paperless system. Google Calendar was later integrated to help with assignment due dates, field trips, and class speakers. Students can be invited to classrooms through the institution's database, through a private code that can then be added in the student's user interface or automatically imported from a school domain Each class created with Google Classroom creates a separate folder in the respective user's Google Drive, where the student can submit work to be graded by a teacher.

Udemy serves as a platform that allows instructors to build online courses on topics of their choosing. Using Udemy's course development tools they can upload video, PowerPoint presentations, PDFs, audio, zip files and live classes to create courses.
Instructors can also engage and interact with users via online discussion boards.

Courses are offered across a breadth of categories, including business and entrepreneurship, academics, the arts, health and fitness, language, music, and technology. Most classes are in practical subjects such as Excel software or using an iPhone camera. Udemy also offers Udemy for Business, enabling businesses access to a targeted suite of over 2,000 training courses on topics from digital marketing tactics to office productivity, design, management, programming, and more. With Udemy for Business, organizations can also create custom learning portals for corporate training.

UDACITY

is a for-profit educational organization founded by Sebastian Thrun, David Stavens, and Mike Sokolsky offering massive open online courses.

According to Thrun, the origin of the name Udacity comes from the company's desire to be "audacious for you, the student While it originally focused on offering university-style courses, it now focuses more on vocational courses for professionals.

Functional Requirements:

defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

1. create post
2. comments on the posts
3. like or dislike posts
4. create course room
5. enroll in a course
6. chatting
7. searching

Non-Functional Requirements:

is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

1- **Usability**: Study Space performs the specific task easily .in other words, it shows the ability of the software to perform tasks. In usability, the software is used by specific consumers to achieve the specific objective with effectiveness, efficiency, and satisfaction.

2- **Reliability:** Study Space works for a long time without any problem and can easily trust the system.

There are two terms related to the software reliability:

Fault: Fault means a defect in the software like a bug in the code which can cause failure in the software.

Failure: Failure means that when the behavior of the software is not same as the specified one.

3- **Maintainability**: It's about code quality how easy for someone else to read the code and maintain application by someone else and the code quality is not good it cannot maintain by someone other so during the process must keep in mind the application must be read or maintained by someone else instead of application maker.

4- **Scalability:** Scalability requirements describe how the system must grow without negative influence on its performance. This means serving more users, processing more data, and doing more transactions. Scalability has both hardware and software implications.

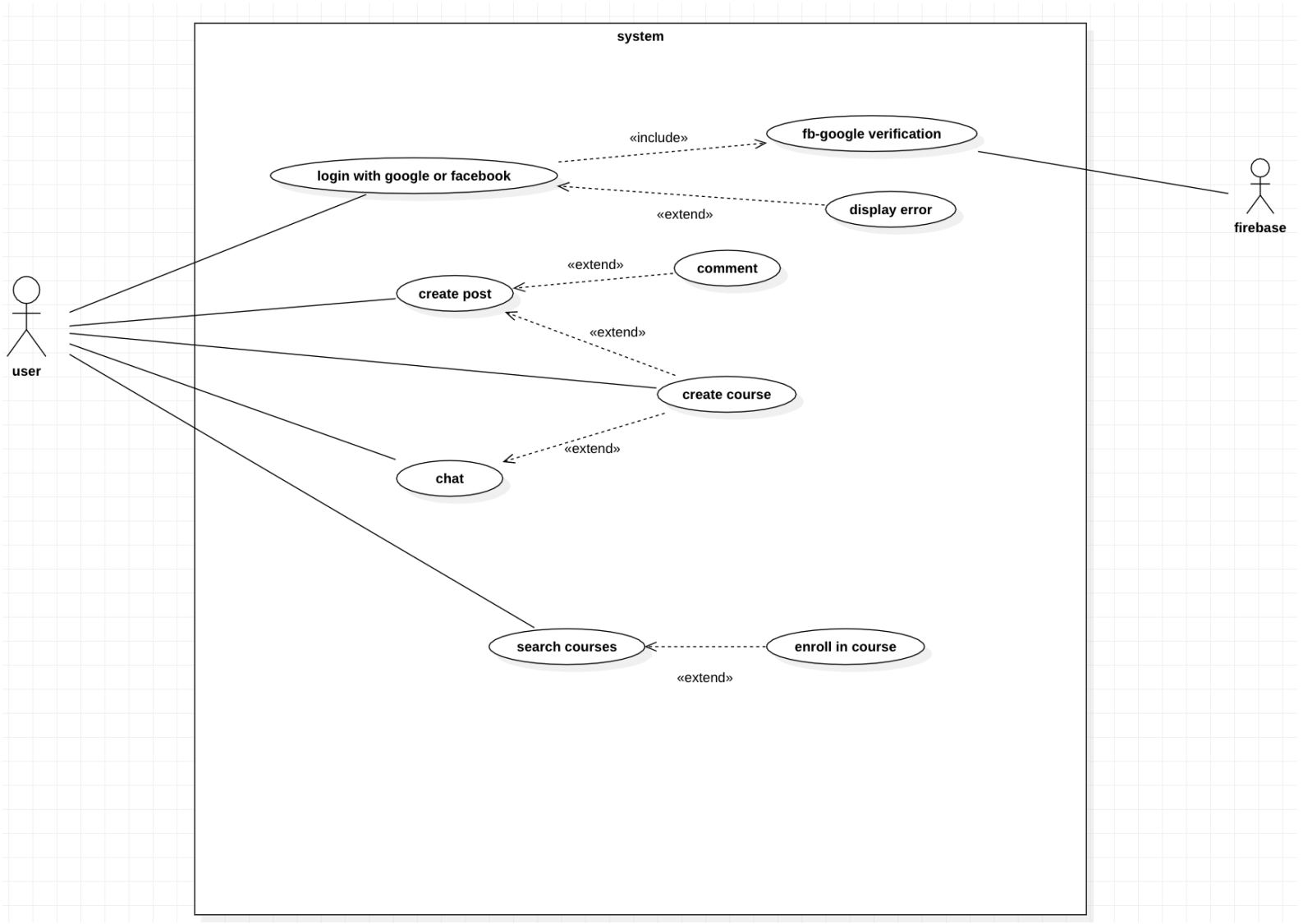
5- **Availability**: Availability is gauged by the period of time that the system's functionality and services are available for use with all operations.

6- **Security:** Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users' roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information.

Use case:

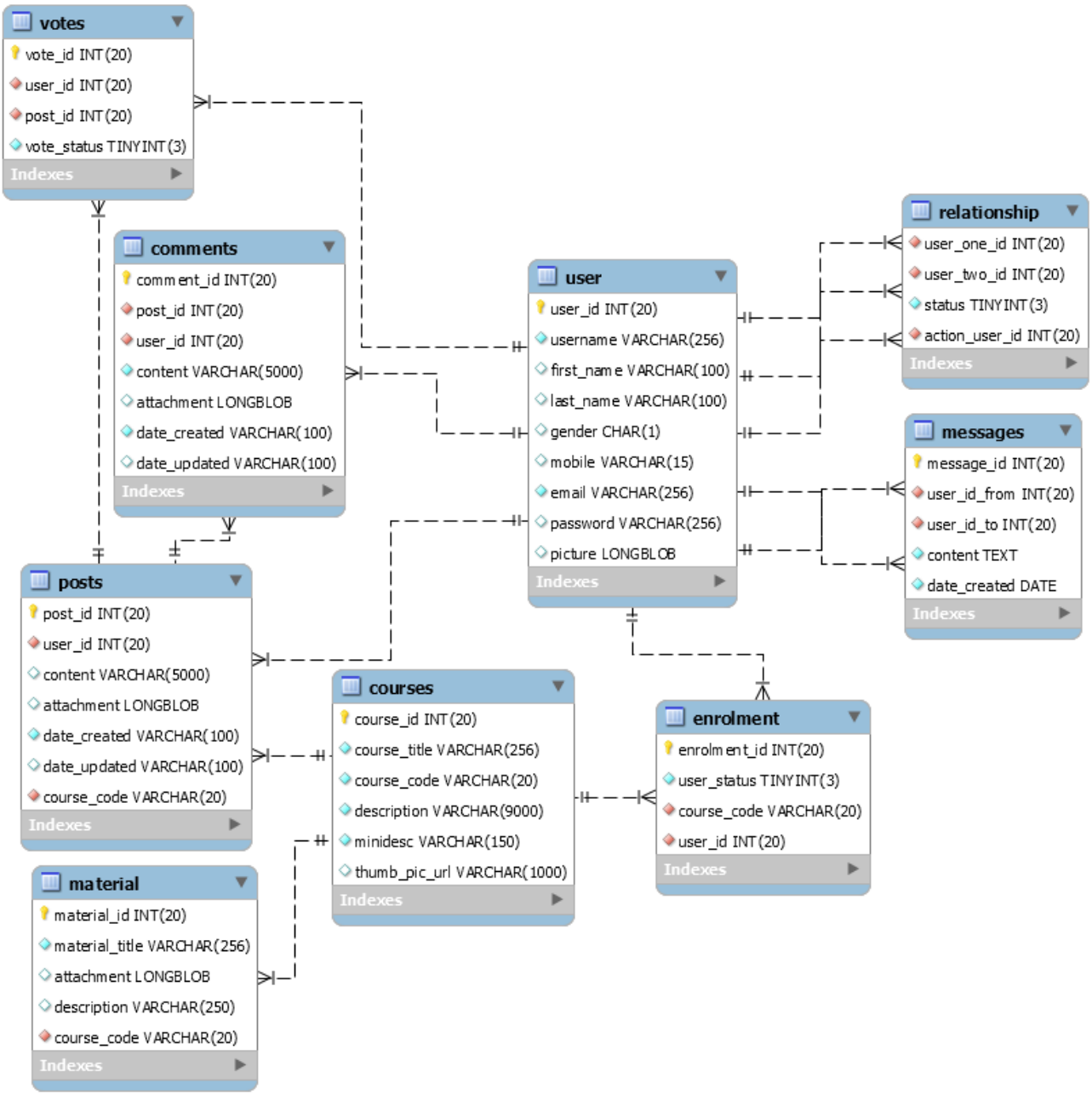
A use case diagram is a graphic depiction of the interactions among the elements of a system.
A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated.
Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

Why Make Use Case Diagrams?
Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities.
They also help identify any internal or external factors that may influence the system and should be taken into consideration.
They provide a good high-level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented


system
fb-google verification
«include»
login with google or facebook
display error
«extend»
firebase
«extend»
comment
create post
«extend»
user
create course
«extend»
chat
search courses
enroll in course
«extend»

ERD

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.
ER diagrams are used to sketch out the design of a database.



Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

Purpose of Class Diagrams

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.
The purpose of the class diagram can be summarized as –
Analysis and design of the static view of an application.
Describe responsibilities of a system.
Base for component and deployment diagrams.
Forward and reverse engineering.

How to Draw a Class Diagram?

Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram.
Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top-level view.
Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represents the whole system.

Screenshots for app code:

1- API Package

API packages handle incoming and outgoing data to/from the application without touching the code itself, it can deal with the server directly.
Simply you can call getData () or createData (), put your parameters and you can easily send and receive data from the server

```
class CourseModel {
  String course_title, course_code, description, minidesc, thumb_pic_url;
 // int course_id, user_id;
  var course_image;

  String firebaseEmail;
  CourseModel({
    this.course_code,
 //   this.course_id,
    this.course_title,
    this.description,
    this.minidesc,
    this.thumb_pic_url,
 //   this.user_id,
    this.course_image,
    this.firebaseEmail
  });

  factory CourseModel.fromJson(Map<String, dynamic> json) => CourseModel(
        course_title: json["course_title"],
        minidesc: json["minidesc"],
        description: json["description"],
        thumb_pic_url: json["thumb_pic_url"],
   //     course_id: json["course_id"],
        course_code: json["course_code"],
   //     user_id: json["user_id"],
        course_image: json["course_image"],
      );
}


List<CourseModel> allPostsFromJson(String str) {
  final jsonData = course_model.json.decode(str);
  return List<CourseModel>.from(jsonData.map((x) => CourseModel.fromJson(x)));
}
```

```dart
class PostModel {
  int postId;
  int userId;
  String content;
  dynamic attachment;
  String dateCreated;
  dynamic dateUpdated;
  String courseCode;
  String username;
  var picture;

  PostModel(
      {this.postId,
      this.userId,
      this.content,
      this.attachment,
      this.dateCreated,
      this.dateUpdated,
      this.courseCode,
      this.username,
      this.picture});

  factory PostModel.fromJson(Map<String, dynamic> json) => PostModel(
        //  postId: json["post_id"],
        //  userId: json["user_id"],
        content: json["content"],
        attachment: json["attachment"],
        dateCreated: json["date_created"],
        dateUpdated: json["date_updated"],
        picture: json["picture"],
        username: json["username"],
      );
}
```

```dart
class NewsFeedModel {
    String a;
    String postId;
    String userId;
    String content;
     String username;
    dynamic attachment;
    String dateCreated;
    dynamic dateUpdated;
    String courseCode;
    String courseTitle;

    NewsFeedModel({
        this.postId,
        this.userId,
        this.content,
        this.username,
        this.attachment,
        this.dateCreated,
        this.dateUpdated,
        this.courseCode,
        this.courseTitle,
    });

    factory NewsFeedModel.fromRawJson(String str) => NewsFeedModel.fromJson(news.json.decode(str));

    String toRawJson() => news.json.encode(toJson());

    factory NewsFeedModel.fromJson(Map<String, dynamic> json) => NewsFeedModel(
        postId: json["post_id"],
        userId: json["user_id"],
        username: json["username"],
        content: json["content"],
        attachment: json["attachment"],
        dateCreated: json["date_created"],
        dateUpdated: json["date_updated"],
        courseCode: json["course_code"],
        courseTitle: json["course_title"],
    );
    Map<String, dynamic> toJson() => {
        "post_id": postId,
        "user_id": userId,
        "username":username,
        "content": content,
        "attachment": attachment,
        "date_created": dateCreated,
        "date_updated": dateUpdated,
        "course_code": courseCode,
        "course_title": courseTitle,
    };
}
```

```dart
class User {
  String username;
  String firstName;
  String lastName;
  String gender;
  String mobile;
  String email;
  String password;
  dynamic picture;

  User({
    this.username,
    this.firstName,
    this.lastName,
    this.gender,
    this.mobile,
    this.email,
    this.password,
    this.picture,
  });

  factory User.fromRawJson(String str) => User.fromJson(user.json.decode(str));

  String toRawJson() => user.json.encode(toJson());

  factory User.fromJson(Map<String, dynamic> json) => User(
        username:  json["username"],
        firstName: json["first_name"] == null ? null : json["first_name"],
        lastName: json["last_name"] == null ? null : json["last_name"],
        gender: json["gender"] == null ? null : json["gender"],
        mobile: json["mobile"] == null ? null : json["mobile"],
        email: json["email"],
        password: json["password"] == null ? null : json["password"],
        picture: json["picture"],
      );

  Map<String, dynamic> toJson() => {
        "username":  username,
        "first_name": firstName == null ? null : firstName,
        "last_name": lastName == null ? null : lastName,
        "gender": gender == null ? null : gender,
        "mobile": mobile == null ? null : mobile,
        "email": email ,
        "password": password == null ? null : password,
        "picture": picture,
      };
}
```

1- Localization Package

Localization package handle the language of our applications very easily by detecting the language of user's cell phones, users also have the choice to change it in the settings and choose the preferred language

```
class I18n implements WidgetsLocalizations {
  const I18n();

  static const GeneratedLocalizationsDelegate delegate =
    const GeneratedLocalizationsDelegate();

  static I18n of(BuildContext context) =>
    Localizations.of<I18n>(context, WidgetsLocalizations);

  @override
  TextDirection get textDirection => TextDirection.ltr;

  String get btnHome => "Home";
  String get btnSearch => "Search";
  String get btnMore => "More";
  String get btnProfile => "Profile";
}
```

```
class _I18n_en_US extends I18n {
  const _I18n_en_US();

  @override
  TextDirection get textDirection => TextDirection.ltr;
}
```

```dart
class _I18n_ar_EG extends I18n {
  const _I18n_ar_EG();

  @override
  String get btnHome => "الرئيسية";
  @override
  String get btnSearch => "بحث";
  @override
  String get btnMore => "المزيد";
  @override
  String get btnProfile => "الصفحة الشخصية";

  @override
  TextDirection get textDirection => TextDirection.rtl;
}
```

```dart
class GeneratedLocalizationsDelegate extends LocalizationsDelegate<WidgetsLocalizations> {
  const GeneratedLocalizationsDelegate();

  List<Locale> get supportedLocales {
    return const <Locale>[
      const Locale("en", "US"),
      const Locale("ar", "EG"),
      const Locale("tr", "TR")
    ];
  }
```

1- Course Package

Course page have list of all course's users enrolled, created or even favorited it
Course page contain tabs that can navigate to post, chat, attachments and create new post

```
class _SliverAppBarDelegate extends SliverPersistentHeaderDelegate {
  _SliverAppBarDelegate(this._tabBar);

  final TabBar _tabBar;

  @override
  double get minExtent => _tabBar.preferredSize.height;

  @override
  double get maxExtent => _tabBar.preferredSize.height;

  @override
  Widget build(
      BuildContext context, double shrinkOffset, bool overlapsContent) {
    return new Container(
      child: _tabBar,
    );
  }

  @override
  bool shouldRebuild(_SliverAppBarDelegate oldDelegate) {
    return false;
  }
}
```

```
class _CoursePageState extends State<CoursePage> {
  @override
  Widget build(BuildContext context) {
    print("course id check>>>>>>>>>>>>>>>>" + widget.courseTitle.toString());

    return Scaffold(
      floatingActionButton: FloatingActionButton(
        onPressed: () async {
          _asyncInputDialog(context);
        },
        child: Icon(Icons.add),
      ),
      body: DefaultTabController(
        length: 3,
        child: NestedScrollView(
            headerSliverBuilder:
                (BuildContext context, bool innerBoxIsScrolled) {
              return <Widget>[
                SliverAppBar(
                  actions: <Widget>[
                    IconButton(
                      onPressed: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                              builder: (BuildContext context) => ChatScreen(
                                    courseName: widget.courseTitle,
                                    name: widget.name,
                                    email: widget.email,
                                    photo: widget.photo,
                                  )),
                        );
                      },
                      icon: Icon(Icons.chat_bubble_outline),
                    )
                  ],
                  expandedHeight: 200.0,
                  floating: true,
                  pinned: true,
                  flexibleSpace: FlexibleSpaceBar(
                      centerTitle: true,
                      title: Text(widget.courseTitle,
                          style: TextStyle(
                            color: Colors.white,
                            fontSize: 16.0,
                          )),
                      background: Image.network(
                        "https://www.thedroidsonroids.com/wp-content/uploads/2019/01/flutter_blog-
750x400.png",
                        fit: BoxFit.cover,
                      )),
                ),
```

```
              SliverPersistentHeader(
                pinned: true,
                floating: true,
                delegate: _SliverAppBarDelegate(
                  TabBar(
                    labelColor: Colors.black87,
                    unselectedLabelColor: Colors.grey,
                    onTap: (int newindex) {
                      switch (newindex) {
                        case 0:
                          break;
                        default:
                      }
                    },
                    tabs: [
                      Tab(
                        icon: Icon(Icons.home),
                        text: "Feeds",
                      ),
                      Tab(
                          icon: Icon(Icons.chat_bubble_outline),
                          text: "Chat"),
                      Tab(icon: Icon(Icons.cloud_download), text: "Material"),
                    ],
                  ),
                ),
              ),
            ];
          },
          body: TabBarView(
            children: <Widget>[
              Container(
                width: MediaQuery.of(context).size.width,
                height: MediaQuery.of(context).size.height,
                color: Colors.white,
                child: Posts(course_code: widget.course_code),
              ),
              //Center(child: Text('Pos// tss are here'),),
              Container(
                width: MediaQuery.of(context).size.width,
                height: MediaQuery.of(context).size.height,
                color: Colors.white,
                child: Center(child: Text("TODO: Remove This Shit")),
              ),
              Container(
                width: MediaQuery.of(context).size.width,
                height: MediaQuery.of(context).size.height,
                color: Colors.white,
                child: Center(
                  child: Text('Downloading and Reviewing material'),
                ),
              ),
            ],
          )),
      ),
    );
  }
}
```

1- Chat package
Users can talk to each other send attachments, images and media content based on the course you are in because chats shown depends on the course name

```
class _ChatScreenState extends State<ChatScreen> {
  final ContactPicker _contactPicker = ContactPicker();

  // File _image;
  var i = 1;

  final TextEditingController _msg = TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.courseName),
        actions: <Widget>[
          IconButton(
            icon: Icon(Icons.attach_file),
            onPressed: () {
              _showDialog();
            },
          ),
          IconButton(
            icon: Icon(Icons.person),
            onPressed: () {
              //      profile(context);
            },
          ),
        ],
      ),
      body: Column(
        children: <Widget>[
          Flexible(
            child: FirebaseAnimatedList(
              defaultChild: CircularProgressIndicator(),
              query: omar(),
              sort: (a, b) => b.key.compareTo(a.key),
              padding: EdgeInsets.all(8.0),
              reverse: true,
              itemBuilder:
                  (_, DataSnapshot snapshot, Animation<double> animation, int) {
                print(snapshot.value['name']);
                return snapshot.value['senderEmail'] != widget.email
                    ? ChatMessage(
                        text: snapshot.value['text'],
                        email: snapshot.value['senderEmail'],
                        pic: snapshot.value['pic'],
                        name: snapshot.value['name'],
                        e: widget.email,
                        type: snapshot.value['type'],
                        url: snapshot.value['url'],
                      )
```

```
                    : Container(

                        child: ChatMessageRight(
                          text: snapshot.value['text'],
                          email: snapshot.value['senderEmail'],
                          pic: snapshot.value['pic'],
                          name: snapshot.value['name'],
                          e: widget.email,
                          type: snapshot.value['type'],
                          url: snapshot.value['url'],
                        ),

                      );
                },
              ),
            ),
            Divider(height: 2.0),
            Container(
              padding: EdgeInsets.all(10.0),
              child: Row(
                children: <Widget>[
                  IconButton(
                      icon: Icon(Icons.camera, color: Colors.blue),
                      onPressed: () async {
                        getImage();
                      }),
                  Flexible(
                    child: TextField(
                      textInputAction: TextInputAction.newline,
                      decoration: InputDecoration(hintText: 'enter message!'),
                      controller: _msg,
                    ),
                  ),
                  IconButton(
                      icon: Icon(
                        Icons.send,
                        color: Colors.blue,
                      ),
                      onPressed: () {
                        sendMsg(_msg.text, "text", "text");
                        _msg.clear();
                        //
                      })
                ],
              ),
            ),
          ],
        ),
      );
    }
}
// single message template--------------------//
```

1- Post Package

Post package contain all the users posts in clouding upvote, downvote and comments based on the course you are in because posts shown depends on the course name

```
class _PostsState extends State<Posts> {
  @override
  Widget build(BuildContext context) {
    return isLoading
        ? Center(
            child: CircularProgressIndicator(
            backgroundColor: Colors.red,
          ))
        : ListView.builder(
            itemCount: posts.length,
            itemBuilder: (BuildContext context, int index) {
              return Post(
                picUrl: posts[index].picture,
                body: posts[index].content,
                numberOflikes: 12,
                numberOfComments: 12,
                date: posts[index].dateCreated,
                //     profilePicUrl: posts[index].profilePicUrl,
                userName: posts[index].username,
                containedCourseName: "course for now",
              );
            });
  }
}
```

2- More package

More package it's just like 'drawer' but we added it to our navigation bar
More package has some additional page like profile, my courses, notes and settings

```
class MoreListTiles extends StatelessWidget {
  MoreListTiles({
    this.icon,
    this.title,
    this.tileOnTap,
    Key key,
  }) : super(key: key);
  final IconData icon;
  final String title;
  final Function tileOnTap;

  @override
  Widget build(BuildContext context) {
    return ListTile(
      title: Padding(
        padding: const EdgeInsets.symmetric(vertical: 10.0),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Icon(icon),
            SizedBox(
              width: 5,
            ),
            Text(title),
          ],
        ),
      ),
      onTap: tileOnTap,
    );
  }
}
```

```dart
class MorePage extends StatelessWidget {
  final String name, email, photo;

  MorePage({this.name, this.photo, this.email});
  @override
  Widget build(BuildContext context) {
    var cntx = context;
    return SafeArea(
      child: Scaffold(
        body: CustomScrollView(
          slivers: <Widget>[
            SliverAppBar(
              title: Text(email),
              pinned: true,
              expandedHeight: 200,
              flexibleSpace: FlexibleSpaceBar(
                title: Text(
                  name,
                  style: TextStyle(shadows: [
                    Shadow(
                        blurRadius: 10.0,
                        color: Colors.black45,
                        offset: Offset(3, 3)),
                  ]),
                ),
                background: Image.network(
                  photo,
                  fit: BoxFit.cover,
                ),
              ),
              actions: <Widget>[],
            ),
            SliverList(
              delegate: SliverChildListDelegate([
                Container(
                  padding: EdgeInsets.only(top: 25),
                  width: MediaQuery.of(context).size.width,
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.center,
                    children: <Widget>[
                      MoreListTiles(
                          title: 'Profile',
                          icon: Icons.person_outline,
                          tileOnTap: () {
                            Navigator.of(context).pushNamed('/profile');
                          }),
                      MoreListTiles(
                        title: 'Friends',
                        icon: Icons.people_outline,
                        tileOnTap: () {
                          Navigator.pushNamed(context, '/friends');
                        },
                      ),
                      MoreListTiles(title: 'Notes', icon: Icons.note),
                      MoreListTiles(title: 'Groups', icon: Icons.group),
                      MoreListTiles(
                        title: 'Courses',
                        icon: Icons.library_books,
                        tileOnTap: () {
                          Navigator.push(
                            context,
                            MaterialPageRoute(
                                builder: (BuildContext context) =>
                                    MyCoursesPage(
                                      firebaseEmail: email,
                                      firebaseName: name,
                                      firebasephoto: photo,
                                    )),
                          );
                        },
                      ),
```

```
                    MoreListTiles(
                        title: 'Settings',
                        icon: Icons.settings,
                        tileOnTap: () {
                          Navigator.of(context).pushNamed('/settings');
                        }),
                    MoreListTiles(
                        title: 'Sign Out',
                        icon: Icons.exit_to_app,
                        tileOnTap: () {
                          showDialog(
                              context: context,
                              builder: (BuildContext context) => Dialog(
                                    elevation: 5,
                                    insetAnimationDuration:
                                        Duration(seconds: 2),
                                    child: Column(
                                      mainAxisSize: MainAxisSize.min,
                                      mainAxisAlignment:
                                          MainAxisAlignment.center,
                                      children: <Widget>[
                                        SizedBox(height: 30),
                                        Text(
                                            'Do you really want to Sign Out ?'),
                                        SizedBox(height: 30),
                                        Row(
                                          mainAxisSize: MainAxisSize.max,
                                          mainAxisAlignment:
                                              MainAxisAlignment.center,
                                          children: <Widget>[
                                            RaisedButton(
                                              child: Text('YES'),
                                              onPressed: () {
                                                Navigator.of(context).pop();
                                                Provider.of<UserRepository>(
                                                        cntx)
                                                    .signOut();
                                              },
                                            ),
                                            SizedBox(width: 10),
                                            RaisedButton(
                                              child: Text('No'),
                                              onPressed: () {
                                                Navigator.of(context).pop();
                                              },
                                            )
                                          ],
                                        ),
                                        SizedBox(height: 30),
                                      ],
                                    ),
                                  ));
                        }),
                  ],
                ),
              ),
            ),
          ]),
        )
      ],
    ),
  ),
);
  }
}
```

3- Platforms Package

Platform package handle the change in iOS/Android design.
Android platform use material design and iOS platforms uses Cupertino design
And platforms package can handle these different very good

```dart
abstract class PlatformWidget<I extends Widget,A extends Widget> extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    if(Platform.isAndroid) {
      return createAndroidWidget(context);
    } else if (Platform.isIOS) {
      return createIosWidget(context);
    }
    return new Container();
  }

  I createIosWidget(BuildContext context);
  A createAndroidWidget(BuildContext context);

}
```

```dart
class PlatformSwitch extends PlatformWidget<CupertinoSwitch, Switch> {
  final onChanged;
  final value;

  PlatformSwitch({this.onChanged, this.value});

  @override
  Switch createAndroidWidget(BuildContext context) {
    return  Switch(
      onChanged: onChanged,
      value: value,
    );
  }

  @override
  CupertinoSwitch createIosWidget(BuildContext context) {
    return CupertinoSwitch(
      onChanged: onChanged,
      value: value,
    );
  }
}
```

```dart
class PlatformScaffold extends PlatformWidget<CupertinoPageScaffold, Scaffold> {
  final Widget child;
  final cupertinoNavigationBar;
  final appBar;
  PlatformScaffold({this.child,this.cupertinoNavigationBar,this.appBar});

  @override
  Scaffold createAndroidWidget(BuildContext context) {
    return  Scaffold(
      appBar: appBar,
      body: child,
    );
  }

  @override
  CupertinoPageScaffold createIosWidget(BuildContext context) {
    return CupertinoPageScaffold(
      navigationBar: cupertinoNavigationBar,
      child: child,
    );
  }
}
```

```dart
class PlatformButton extends PlatformWidget<CupertinoButton, RaisedButton> {
  final VoidCallback onPressed;
  final Widget child;

  PlatformButton({this.child, this.onPressed});

  @override
  RaisedButton createAndroidWidget(BuildContext context) {
    return  RaisedButton(
      child: child,
      onPressed: onPressed,
    );
  }

  @override
  CupertinoButton createIosWidget(BuildContext context) {
    return  CupertinoButton(
      child: child,
      onPressed: onPressed,
      color: Colors.blue,
    );
  }
}
```

4- Login Package

With firebase authentication the login package became more secure and support fast and efficient login using Facebook/Google

```
class _LoginPageState extends State<LoginPage> {
  @override
  Widget build(BuildContext context) {
    final user = Provider.of<UserRepository>(context);

    double panelCollapsedHeight = MediaQuery.of(context).size.height * .08;
    double panelExpandedHeight = MediaQuery.of(context).size.height * .3;

    PanelController _pc = PanelController();

    return Scaffold(
      key: _key,
      body: SlidingUpPanel(
        body: Stack(
          fit: StackFit.expand,
          children: <Widget>[
            Image.asset('assets/uni.jpg', fit: BoxFit.cover),
            Container(
              width: MediaQuery.of(context).size.width,
              height: MediaQuery.of(context).size.height,
              color: Colors.red.withOpacity(.3),
            ),
            Wrap(
                runAlignment: WrapAlignment.center,
                direction: Axis.horizontal,
                alignment: WrapAlignment.spaceEvenly,
                crossAxisAlignment: WrapCrossAlignment.start,
                children: <Widget>[
                  Text(
                    'Study Space',
                    textAlign: TextAlign.center,
                    style: TextStyle(
                        color: Colors.white,
                        fontSize: panelCollapsedHeight * .7,
                        fontWeight: FontWeight.bold,
                        letterSpacing: 2),
                  ),
                  Text(
                    'A place where you BELONG',
                    textAlign: TextAlign.center,
                    style: TextStyle(
                        color: Colors.white,
                        fontSize: panelCollapsedHeight * .24,
                        fontWeight: FontWeight.w400,
                        letterSpacing: 2),
                  ),
                ]),
          ],
        ),
```

```
      controller: _pc,
      collapsed: GestureDetector(
        onTap: () => _pc.isPanelClosed() ? _pc.open() : null,
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Icon(
              Icons.arrow_drop_up,
              size: panelCollapsedHeight * .5,
              color: Colors.black12,
            ),
            SizedBox(width: 30),
            Text(
              'Sign in',
              style: TextStyle(
                  color: Colors.black54,
                  fontSize: panelCollapsedHeight * .22),
            ),
            SizedBox(width: 30),
            Icon(
              Icons.arrow_drop_up,
              size: panelCollapsedHeight * .3,
              color: Colors.black12,
            ),
          ],
        ),
      ),
      panel: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          mainAxisAlignment: MainAxisAlignment.center,
          mainAxisSize: MainAxisSize.max,
          children: <Widget>[
            user.status == Status.Authenticating
                ? Center(child: CircularProgressIndicator())
                : Padding(
                    padding: EdgeInsets.symmetric(horizontal: 16.0),
                    child: Container(
                      width: MediaQuery.of(context).size.width -
                          MediaQuery.of(context).size.width * .5,
                      height: panelCollapsedHeight,
                      child: GoogleSignInButton(
                        borderRadius: 50,
                        text: "Sign In With Google",
                        onPressed: () async {
                          if (!await user.signInWithGoogle())
                            _key.currentState.showSnackBar(SnackBar(
                              content: Text(
                                  "Something is wrong , Maybe Check your Internet Connection"),
                            ));
                        },
                      ),
                    )),
            FacebookSignInButton(
              borderRadius: 50,
              text: "Sign In With Facebook",
              onPressed: () async {
                user.signInWithFacebook();
              },
            ),
          ]),
      borderRadius: BorderRadius.only(
          topRight: Radius.circular(40), topLeft: Radius.circular(40)),
      maxHeight: panelExpandedHeight,
      minHeight: panelCollapsedHeight,
      parallaxEnabled: true,
      parallaxOffset: .5,
      backdropEnabled: true,
      backdropOpacity: .4,
    ),
  );
}
```

5- Home Package

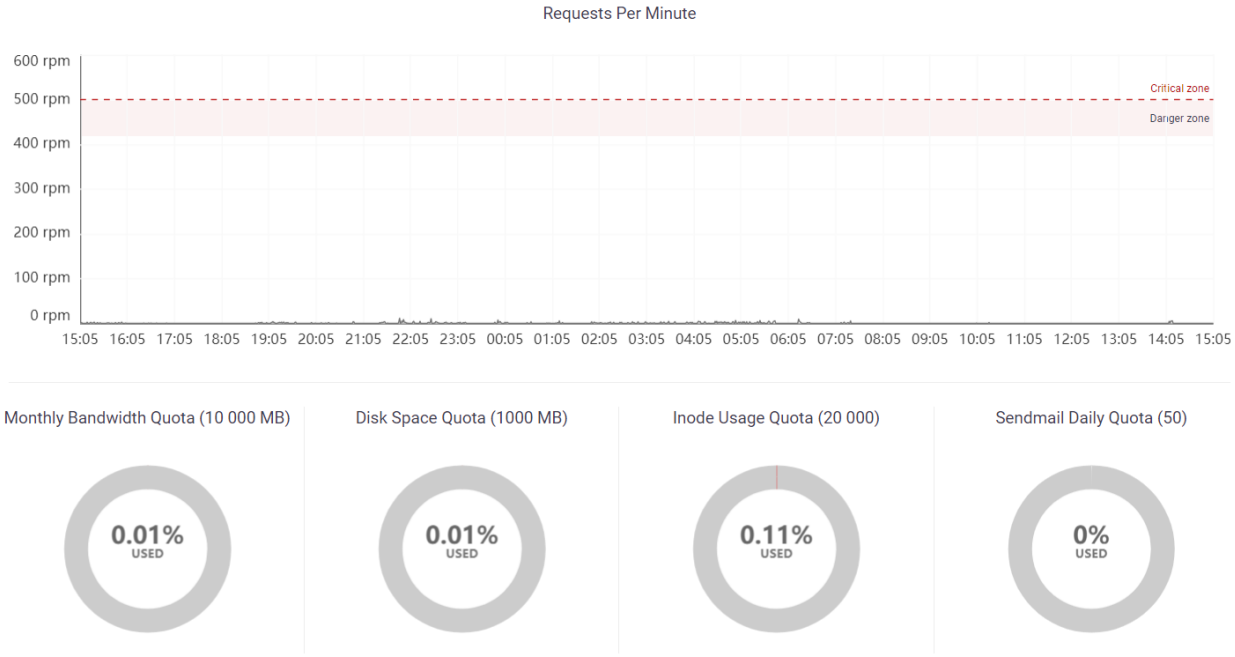
In the home package users can found latest post news and their favorite courses

```
class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: CustomScrollView(
        scrollDirection: Axis.vertical,
        slivers: <Widget>[
          SliverAppBar(
            floating: true,
            pinned: false,
            forceElevated: true,
            centerTitle: true,
            backgroundColor: Theme.of(context).bottomAppBarColor,
            elevation: 2,
            title: Text(
              'Home',
              style: TextStyle(
                  letterSpacing: 3,
                  color: Theme.of(context).primaryColor,
                  fontSize: 20),
            ),
          ),
          SliverList(
            delegate: SliverChildListDelegate([
              SizedBox(height: 20),
              Text(
                '    Favorite ',
                style: TextStyle(
                  fontSize: 30,
                ),
              ),
              SizedBox(height: 20),
              Container(
                height: 300,
                child: PageView.builder(
                    controller: PageController(viewportFraction: .6),
                    scrollDirection: Axis.horizontal,
                    itemCount:  news == null ? 0 : news.length,
                    pageSnapping: true,
                    itemBuilder: (BuildContext context, index) => Container(
                      width: MediaQuery.of(context).size.width * .5,
                      height: 300,
                      child:  Column(
                        children: <Widget>[
                          SizedBox(height: 30),
                          Text(" courseName : " +
                              news[index].courseTitle),
                          SizedBox(height: 10),
                          Text(
                              " userName: " + news[index].username),
                          SizedBox(height: 10),
                          Text(
                              " content  : " + news[index].content),
                          SizedBox(height: 10),
                          Text(" Date created: " +
                              news[index].dateCreated.toString()),
                          SizedBox(height: 30),
                          Divider(),
                        ],
                      ),
                      //  Image.network(
                      //   url[index],
                      //   fit: BoxFit.cover,
                      // ),
                    )),
              ),
```

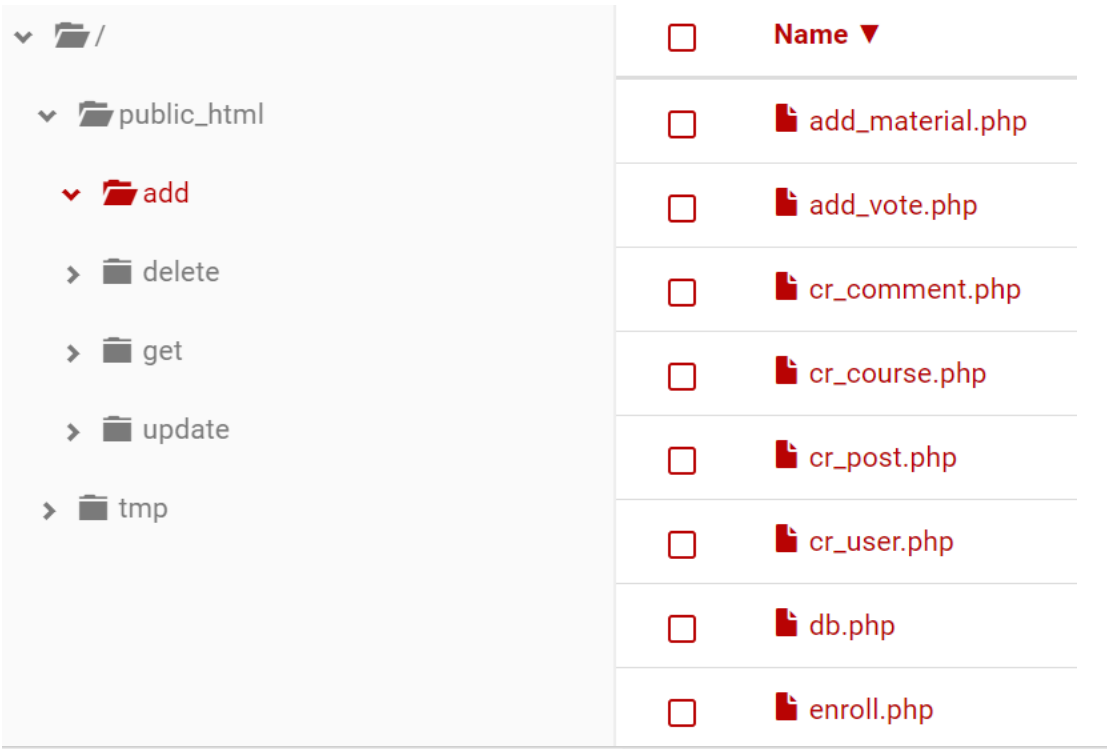
```
              SizedBox(height: 30),
              Text(
                '    professional ',
                style: TextStyle(
                  fontSize: 30,
                ),
              ),
              SizedBox(height: 20),
              Container(
                height: 300,
                child: PageView.builder(
                    controller: PageController(viewportFraction: .6),
                    scrollDirection: Axis.horizontal,
                    itemCount: 4,
                    pageSnapping: true,
                    itemBuilder: (BuildContext context, index) => Stack(
                      fit: StackFit.loose,
                      children: <Widget>[
                        Container(
                            width: MediaQuery.of(context).size.width * .5,
                            height: 300,
                            child: Image.network(
                              url2[index],
                              fit: BoxFit.fitHeight,
                            )),
                      ],
                    )),
              ),
              SizedBox(height: 20),
              Text(
                '    Favorite Tany 3ady',
                style: TextStyle(
                  fontSize: 30,
                ),
              ),
              SizedBox(height: 20),
              Container(
                height: 300,
                child: PageView.builder(
                    controller: PageController(viewportFraction: .6),
                    scrollDirection: Axis.horizontal,
                    itemCount: 4,
                    pageSnapping: true,
                    itemBuilder: (BuildContext context, index) => Stack(
                      fit: StackFit.loose,
                      children: <Widget>[
                        Container(
                          width: MediaQuery.of(context).size.width * .5,
                          height: 300,
                          child: CachedNetworkImage(
                            fit: BoxFit.cover,
                            imageUrl:
                            'https://www.journal-
news.com/rf/image_large/Pub/p9/CmgSharedContent/2018/01/06/Images/GettyImages-155256950-
RtD12E73KjzHTo558WCxcqN-680x383.jpeg',
                            placeholder: (context, url) =>
                                CircularProgressIndicator(),
                            errorWidget: (context, url, error) => Icon(
                              Icons.error_outline,
                              size: 5,
                            ),
                          ),
                        )
                      ],
                    )),
              ),
              SizedBox(
                height: 50,
              )
            ]),
          )
        ],
      ),
    );
  }
}
```

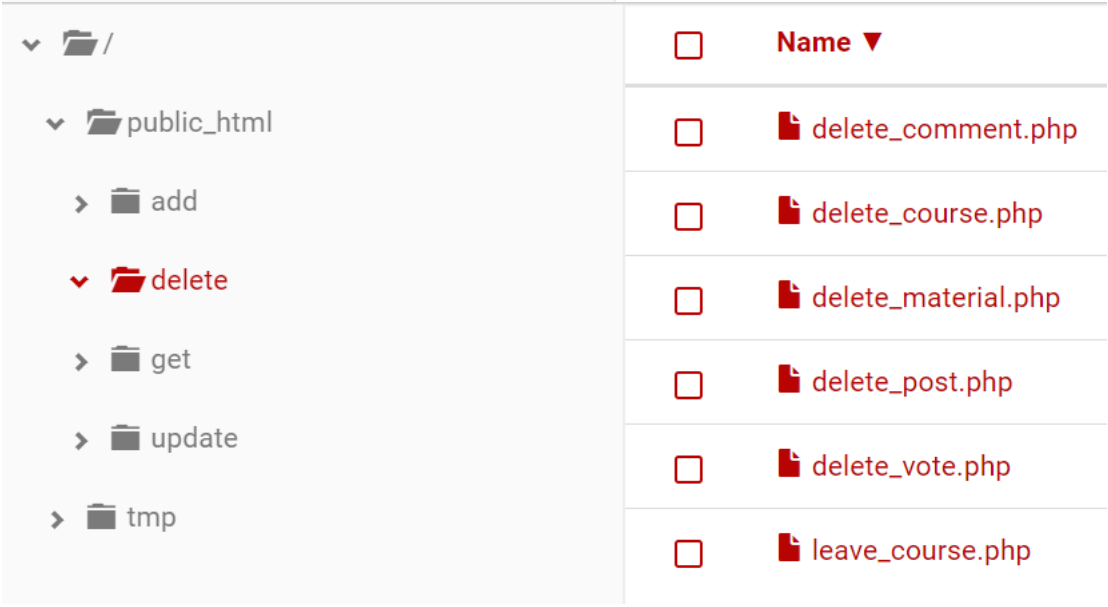
Backend Screenshots:

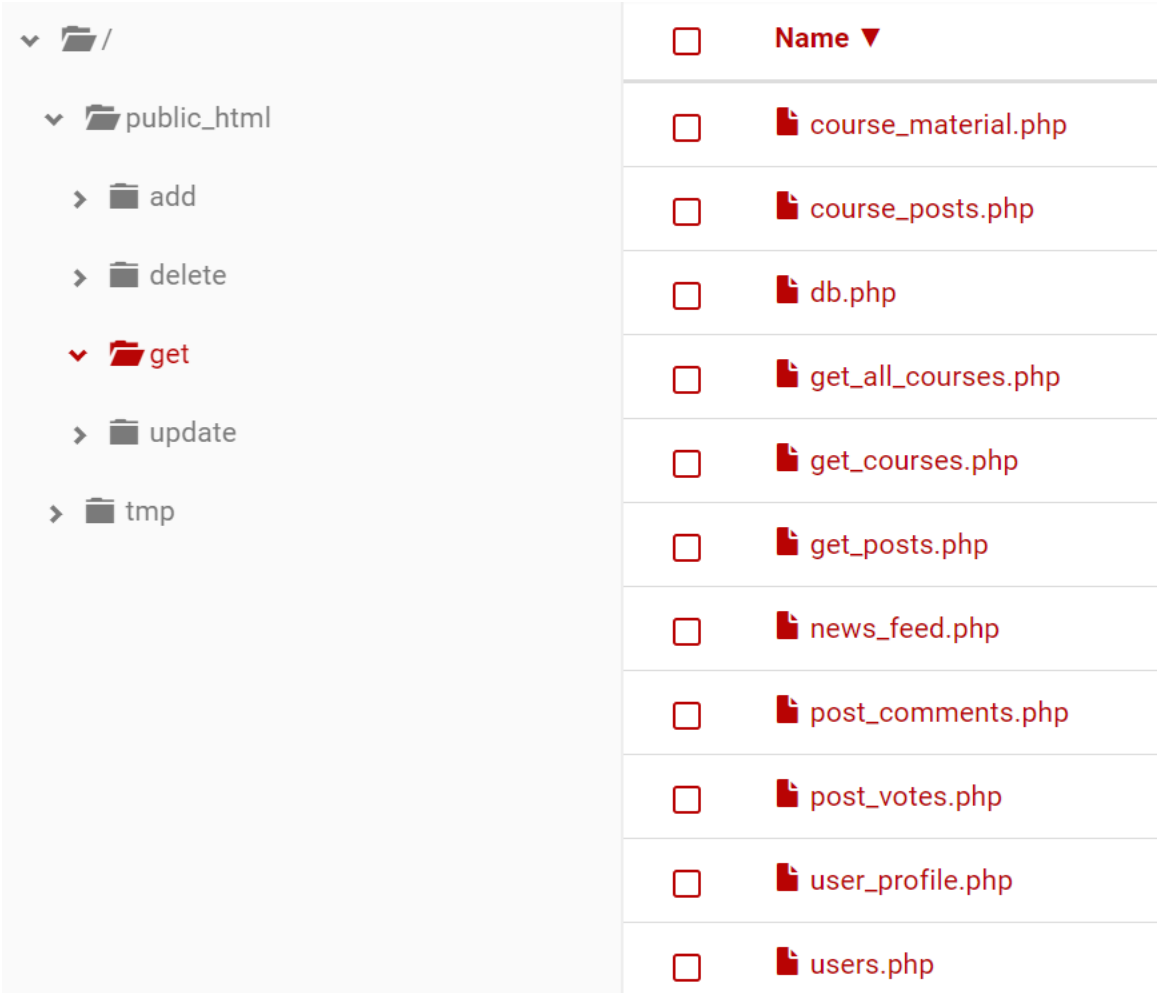
Statistics of StudySpace's Database hosted on 000webhost



PHP Restful API's hosted on 000webhost via it's File Manager to Connect and communicate changes between Studyspace and it's Database


/
public_html
add
delete
get
update
tmp
Name ▼
add_material.php
add_vote.php
cr_comment.php
cr_course.php
cr_post.php
cr_user.php
db.php
enroll.php


/
public_html
add
delete
get
update
tmp
Name ▼
delete_comment.php
delete_course.php
delete_material.php
delete_post.php
delete_vote.php
leave_course.php


/
public_html
add
delete
get
update
tmp
Name ▼
course_material.php
course_posts.php
db.php
get_all_courses.php
get_courses.php
get_posts.php
news_feed.php
post_comments.php
post_votes.php
user_profile.php
users.php

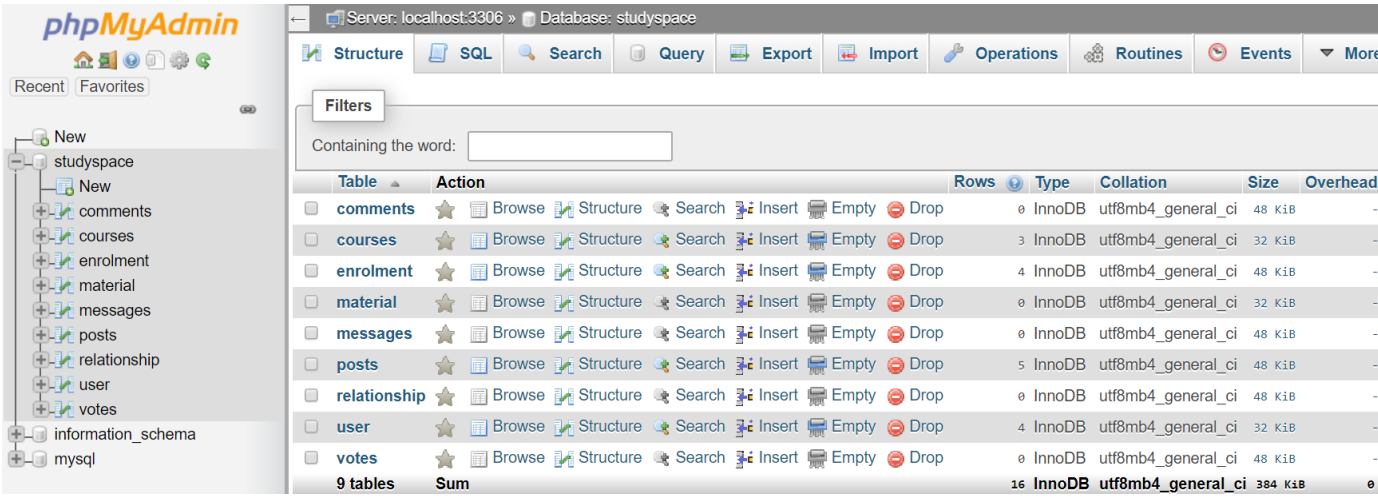
Studyspace’s news feed php file

```php
<?php

require_once('db.php');

$email = $_POST['email'];

$query = "SELECT posts.post_id, posts.user_id, posts.content, posts.attachment, posts.date_created,
posts.date_updated, posts.course_code
,user.username, user.picture, courses.course_title
FROM posts
LEFT JOIN enrolment ON posts.course_code=enrolment.course_code
LEFT JOIN user on posts.user_id=user.user_id
LEFT Join courses on posts.course_code=courses.course_code
where enrolment.user_id= (select user.user_id from user
where user.email = '$email') and enrolment.user_status!=0";

$stm = $db->prepare($query);

$stm -> execute();


$res =array();
while ($row = $stm -> fetch(PDO :: FETCH_ASSOC)){
    $res[]=$row ;
}

echo json_encode($res);

?>
```

Studyspace’s create courses php file

```php
<?php

require_once('db.php');

$course_title = $_POST['course_title'];

$course_code = $_POST['course_code'];

$description =$_POST['description'];

$minidesc = $_POST['minidesc'];



$email = $_POST['email'];

$query = "insert into courses (course_title,course_code,description,minidesc) values
('$course_title','$course_code', '$description','$minidesc' )  ;

 insert into enrolment
(user_id, user_status, course_code) values ((select user.user_id from user where email = '$email'),
'3',
 '$course_code') " ;


$stm = $db -> prepare ($query);


$stm -> execute();
```

000webhosting's phpmyadmin main view of the studyspace database



Data Dictionary

comments

Column	Type	Null	Default	Comments
comment_id *(Primary)*	int(20)	No		
post_id	int(20)	No		
user_id	int(20)	No		
content	varchar(5000)	No		
attachment	longblob	Yes	*NULL*	
date_created	varchar(100)	No		
date_updated	varchar(100)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	comment_id	0	A	No	
post_id	BTREE	No	No	post_id	0	A	No	
user_id	BTREE	No	No	user_id	0	A	No	

courses

Column	Type	Null	Default	Comments
course_id *(Primary)*	int(20)	No		
course_title	varchar(256)	No		
course_code	varchar(20)	No		
description	varchar(9000)	No		
minidesc	varchar(150)	No		
thumb_pic_url	varchar(1000)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	course_id	1	A	No	
course_code	BTREE	Yes	No	course_code	1	A	No	

enrolment

Column	Type	Null	Default	Comments
enrolment_id *(Primary)*	int(20)	No		
user_status	tinyint(3)	No	0	
course_code	varchar(20)	No		
user_id	Int(20)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	enrolment_id	3	A	No	
user_id	BTREE	No	No	user_id	3	A	No	
enrolment_ibfk_3	BTREE	No	No	course_code	3	A	No	

material

Column	Type	Null	Default	Comments
material_id *(Primary)*	int(20)	No		
material_title	varchar(256)	No		
attachment	longblob	Yes	*NULL*	
description	varchar(250)	Yes	*NULL*	
course_code	varchar(20)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	material_id	0	A	No	
material_ibfk_1	BTREE	No	No	course_code	0	A	No	

messages

Column	Type	Null	Default	Comments
message_id *(Primary)*	int(20)	No		
user_id_from	int(20)	No		
user_id_to	int(20)	No		
content	text	No		
date_created	date	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	message_id	0	A	No	
user_id_from	BTREE	No	No	user_id_from	0	A	No	
user_id_to	BTREE	No	No	user_id_to	0	A	No	

posts

Column	Type	Null	Default	Comments
post_id *(Primary)*	int(20)	No		
user_id	int(20)	No		
content	varchar(5000)	Yes	*NULL*	
attachment	longblob	Yes	*NULL*	
date_created	varchar(100)	No		
date_updated	varchar(100)	Yes	*NULL*	
course_code	varchar(20)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	post_id	6	A	No	
user_id	BTREE	No	No	user_id	6	A	No	
posts_ibfk_2	BTREE	No	No	course_code	6	A	No	

relationship

Column	Type	Null	Default	Comments
user_one_id	int(20)	No		
user_two_id	int(20)	No		
status	tinyint(3)	No	0	
action_user_id	int(20)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
unique_users_id	BTREE	Yes	No	user_one_id	0	A	No	
				user_two_id	0	A	No	
user_two_id	BTREE	No	No	user_two_id	0	A	No	
action_user_id	BTREE	No	No	action_user_id	0	A	No	

user

Column	Type	Null	Default	Comments
user_id *(Primary)*	int(20)	No		
username	varchar(256)	No		
first_name	varchar(100)	Yes	*NULL*	
last_name	varchar(100)	Yes	*NULL*	
gender	char(1)	Yes	*NULL*	
mobile	varchar(15)	Yes	*NULL*	
email	varchar(256)	No		
password	varchar(256)	Yes	*NULL*	
picture	longblob	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	user_id	3	A	No	
email	BTREE	Yes	No	email	3	A	No	

votes

Column	Type	Null	Default	Comments
vote_id *(Primary)*	int(20)	No		
user_id	int(20)	No		
post_id	int(20)	No		
vote_status	tinyint(3)	No	0	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	vote_id	0	A	No	
user_id	BTREE	No	No	user_id	0	A	No	
post_id	BTREE	No	No	post_id	0	A	No	

Conclusions:

The StudySpace app presented here can achieve all student passion
and we hope this application help over 1000,000 students next year after publishing the app on both AppStore And Google Play Store.

STUDYSPACE