



HANDS-ON MODERN MOBILE DEV FOR THE OPEN EDX PLATFORM





ARCHITECTURE





CURRENT APPS ARCHITECTURE



~~MVC - MODEL VIEW CONTROLLER~~
MASSIVE VIEWCONTROLLER





AUTHORIZATION



OEXLoginViewController.m:

722 lines

LoginActivity.java + activity_login.xml:

593 lines

OEXRegistrationViewController.m + Swift ext:

906 lines

RegisterActivity.java + activity_register.xml:

870 lines

lines of UI,
Networking,
and Business Logic



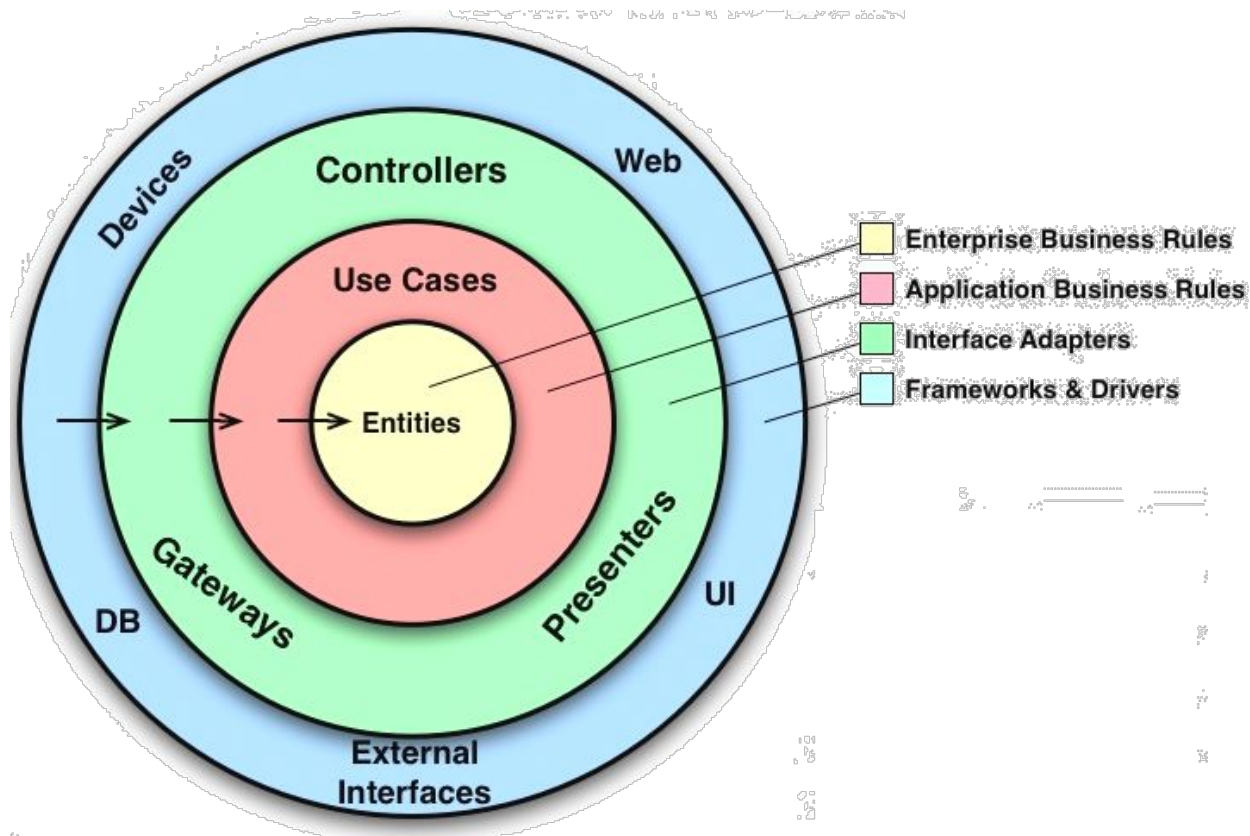


SILVER BULLET



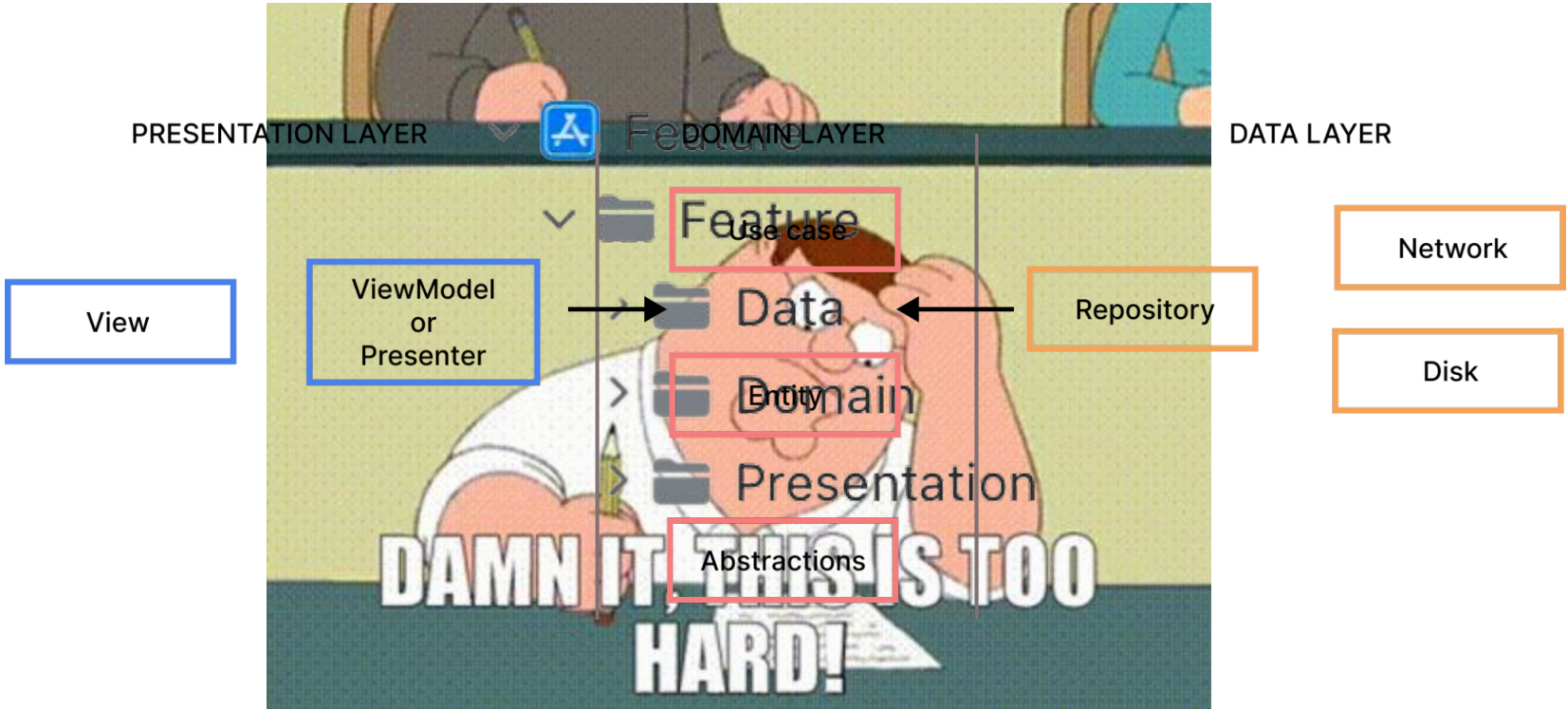


CLEAN ARCHITECTURE BY UNCLE BOB





LAYERS OF ABSTRACTION





PROJECT STRUCTURE





STRUCTURE OF THE CURRENT IOS PROJECT



- edX
 - edX
 - App
 - Source
 - Settings
 - Payment
 - ChromeCast
 - Deep Linking
 - Analytics
 - App Review
 - Application
 - Authentication
 - Base Structures
 - Certificates and Badges
 - Configuration
 - Controller Helpers
 - Course
 - Course Enrollment
 - Custom Styles
 - Data
 - Debug
 - Discussion
 - Download View
 - Front Rear views
 - Helper Views
 - Library Additions
 - Lists
 - Localization
 - Model
 - Offline
 - PassthroughView
 - Push Notifications
 - SocialLogin
 - Swipeable Cell
 - Supporting Files
 - User Profiles
 - Utils
 - VersionUpgrade
 - Videos
 - WhatsNew
 - XIB/Storyboard
 - Tests
 - Resources
 - External Libraries
 - Frameworks





STRUCTURE OF THE CURRENT ANDROID PROJECT



- OpenEdXMobile
 - manifests
 - java
 - org
 - edx
 - mobile
 - authenticatic
 - base
 - core
 - course
 - deeplink
 - discussion
 - event
 - exception
 - extenstion
 - googlecast
 - http
 - inappurcha
 - instrumentat
 - interfaces
 - loader
 - logger
 - model
 - module
 - notifications
 - player
 - profiles
 - receivers
 - repository
 - services
 - social
 - task
 - test
 - third_party
 - user
 - util
 - view
 - viewModel
 - whatsnew
 - wrapper





ONE MORE PAIN 🥲



- view
 - adapters
 - common
 - custom
 - dialog
 - login
 - view_holders
 - AccountActivity
 - AccountFragment
 - AuthenticatedWebViewActivity
 - AuthenticatedWebViewFragment
 - AuthPanelUtils
 - BaseCourseUnitVideoFragment
 - BaseWebViewFragment
 - BulkDownloadFragment
 - CertificateActivity
 - CertificateFragment
 - CourseAnnouncementsActivity
 - CourseAnnouncementsFragment
 - CourseBaseActivity
 - CourseBaseActivity
 - CourseDatesPageFragment
 - CourseDetailActivity
 - CourseDetailFragment
 - CourseDiscussionCommentsActivity
 - CourseDiscussionCommentsFragment
 - CourseDiscussionPostsActivity
 - CourseDiscussionPostsBaseFragment
 - CourseDiscussionPostsSearchFragment
 - CourseDiscussionPostsThreadFragment
 - CourseDiscussionResponsesActivity
 - CourseDiscussionResponsesFragment
 - CourseDiscussionTopicsFragment
 - CourseHandoutActivity
 - CourseHandoutFragment
 - CourseInfoActivity
 - CourseOutlineActivity
 - CourseOutlineFragment
 - CourseTabsDashboardActivity
 - CourseTabsDashboardFragment
 - CourseUnitDiscussionFragment
 - CourseUnitEmptyFragment
 - CourseUnitFragment
 - CourseUnitMobileNotSupportedFragment
 - CourseUnitNavigationActivity
 - CourseUnitOnlyOnYoutubeFragment
 - CourseUnitVideoPlayerFragment
 - CourseUnitWebViewFragment
 - CourseUnitYoutubePlayerFragment
 - CourseUpgradeWebViewActivity
 - CourseUpgradeWebViewFragment
 - CroplImageActivity
 - DiscoveryActivity
 - DiscoveryLaunchActivity
 - DiscoveryLaunchPresenter
 - DiscussionAddCommentActivity
 - DiscussionAddCommentFragment
 - DiscussionAddPostActivity
 - DiscussionAddPostFragment
 - DiscussionAddResponseActivity

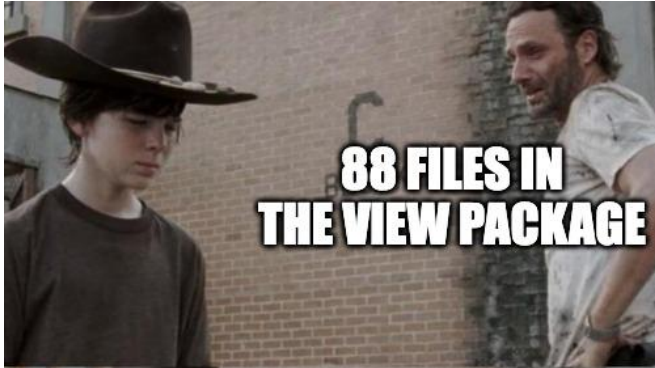


ONE MORE PAIN 🥲



- DiscussionAddResponseFragment
- DownloadListActivity
- EditUserProfileActivity
- EditUserProfileFragment
- FormFieldActivity
- FormFieldSelectFragment
- FormFieldTextAreaFragment
- GroupDiscussionFragment
- ICommonUI
- LaunchActivity
- LockedCourseUnitFragment
- LoginActivity
- MainDashboardActivity
- MainDiscoveryFragment
- MainTabsDashboardFragment
- MyCoursesListFragment
- OfflineSupportBaseActivity
- OfflineSupportBaseFragment
- OfflineSupportUtils
- OnSwipeListener
- PaymentsBannerFragment
- PaymentsInfoActivity
- PaymentsInfoFragment
- Presenter
- PresenterActivity
- PresenterFragment
- ProgramInfoActivity
- ProgramWebViewActivity
- RegisterActivity
- ResourcesFragment
- Router
- SplashActivity
- TabsBaseFragment
- ViewHoldingPresenter
- WebViewDiscoverFragment
- WebViewProgramFragment

ONE MORE PAIN 🥲





ONE MORE SILVER BULLET














Modularization

















IOS:

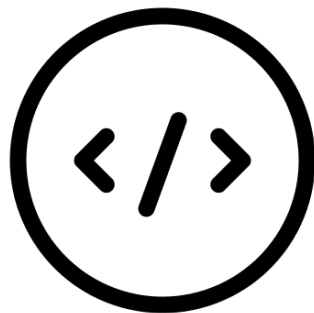
- >  NewEdX
- >  Core
- >  Authorization
- >  Discovery
- >  Dashboard
- >  Profile
- >  Course
- >  Discussion
- >  Pods

ANDROID:

- >  **app**
- >  **auth**
- >  **core**
- >  **course**
- >  **dashboard**
- >  **discovery**
- >  **discussion**
- >  **profile**
- >  Gradle Scripts

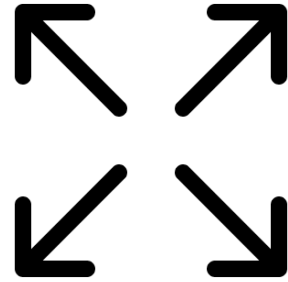
FEATURE:

- ∨  **profile**
 - >  manifests
 - ∨  java
 - ∨  com
 - ∨  raccoongang
 - ∨  profile
 - >  data
 - >  domain
 - ∨  presentation
 - >  delete
 - >  edit
 - >  profile
 - >  settings
 - >  ProfileRouter



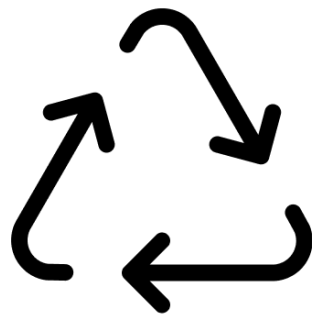
IMPROVED CODE ORGANIZATION





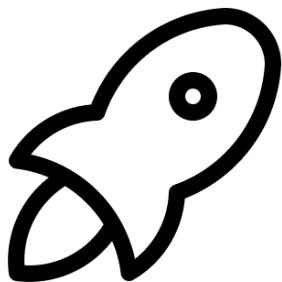
SCALABILITY





REUSABILITY





FASTER BUILD TIME





TECH STACK





IMPERATIVE OR DECLARATIVE UI?



```
class ListTableViewCell: UITableViewCell {
    static let reuseIdentifier = "ListTableViewCell"

    private let titleLabel: UILabel = {
        let label = UILabel()
        label.font = UIFont.boldSystemFont(ofSize: 18)
        label.translatesAutoresizingMaskIntoConstraints = false
        return label
    }()

    private let subtitleLabel: UILabel = {
        let label = UILabel()
        label.font = UIFont.systemFont(ofSize: 14)
        label.translatesAutoresizingMaskIntoConstraints = false
        return label
    }()

    override init(style: UITableViewCell.CellStyle, reuseIdentifier: String?) {
        super.init(style: style, reuseIdentifier: reuseIdentifier)
        contentView.addSubview(titleLabel)
        contentView.addSubview(subtitleLabel)
        setupConstraints()
    }

    required init?(coder: NSCoder) {
        fatalError("init(coder:) has not been implemented")
    }

    private func setupConstraints() {
        NSLayoutConstraint.activate([
            titleLabel.topAnchor.constraint(equalTo: contentView.topAnchor, constant: 16),
            titleLabel.leadingAnchor.constraint(equalTo: contentView.leadingAnchor, constant: 16),
            titleLabel.trailingAnchor.constraint(equalTo: contentView.trailingAnchor, constant: -16),

            subtitleLabel.topAnchor.constraint(equalTo: titleLabel.bottomAnchor, constant: 8),
            subtitleLabel.leadingAnchor.constraint(equalTo: contentView.leadingAnchor, constant: 16),
            subtitleLabel.trailingAnchor.constraint(equalTo: contentView.trailingAnchor, constant: -16),
            subtitleLabel.bottomAnchor.constraint(equalTo: contentView.bottomAnchor, constant: -16)
        ])
    }

    func configure(with item: ListItem) {
        titleLabel.text = item.title
        subtitleLabel.text = item.subtitle
    }
}
```

```
class ViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {
    private let tableView = UITableView()

    private let items = [
        ListItem(title: "Title 1", subtitle: "Subtitle 1"),
        ListItem(title: "Title 2", subtitle: "Subtitle 2"),
        ListItem(title: "Title 3", subtitle: "Subtitle 3")
    ]

    override func viewDidLoad() {
        super.viewDidLoad()

        tableView.dataSource = self
        tableView.delegate = self
        tableView.register(ListTableViewCell.self, forCellReuseIdentifier: ListTableViewCell.reuseIdentifier)
        tableView.translatesAutoresizingMaskIntoConstraints = false
        view.addSubview(tableView)

        NSLayoutConstraint.activate([
            tableView.topAnchor.constraint(equalTo: view.topAnchor),
            tableView.bottomAnchor.constraint(equalTo: view.bottomAnchor),
            tableView.leadingAnchor.constraint(equalTo: view.leadingAnchor),
            tableView.trailingAnchor.constraint(equalTo: view.trailingAnchor)
        ])
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
        return items.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: ListTableViewCell.reuseIdentifier, for: indexPath)
        cell.configure(with: items[indexPath.row])
        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        tableView.deselectRow(at: indexPath, animated: true)
        print("Item tapped: \(items[indexPath.row])")
    }
}
```



SWIFTUI



```
struct ContentView: View {
    let items = [
        ListItem(title: "Title 1", subtitle: "Subtitle 1"),
        ListItem(title: "Title 2", subtitle: "Subtitle 2"),
        ListItem(title: "Title 3", subtitle: "Subtitle 3")
    ]

    var body: some View {
        NavigationView {
            List(items) { item in
                ListItemView(item: item)
                    .onTapGesture {
                        print("Item tapped: \(item)")
                    }
            }
            .navigationTitle("Items")
        }
    }
}
```



JETPACK COMPOSE



```
@Composable
fun ContentView() {
    val items = listOf(
        ListItem( title: "Title 1", subtitle: "Subtitle 1"),
        ListItem( title: "Title 2", subtitle: "Subtitle 2"),
        ListItem( title: "Title 3", subtitle: "Subtitle 3")
    )

    Surface {
        LazyColumn { this: LazyListScope
            items(items) { this: LazyItemScope item ->
                ListItemView(item = item, onClick = {
                    Log.d( tag: "ComposeExample", msg: "Item clicked: $item")
                })
            }
        }
    }
}
```

(top = 8.dp))



THEMING

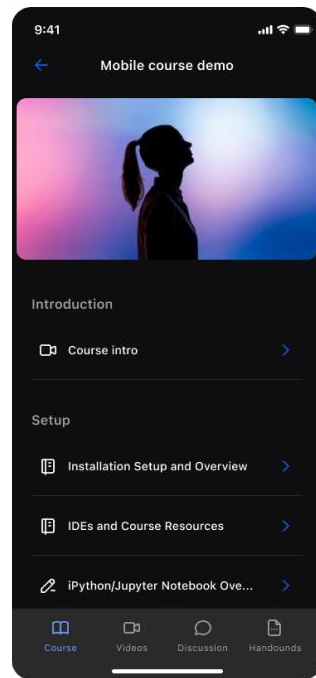
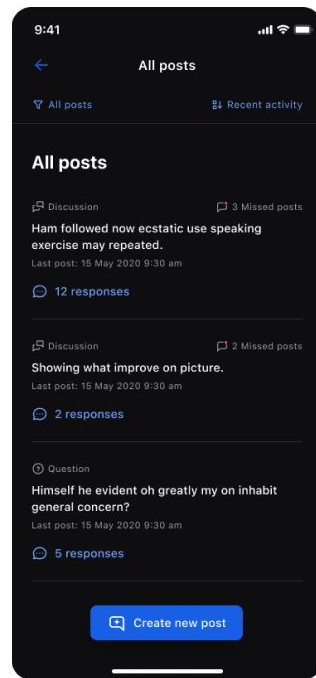
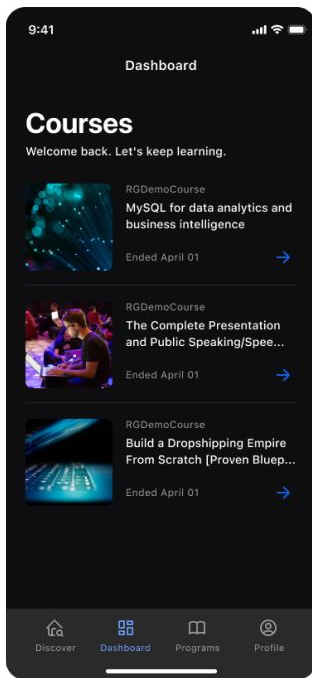
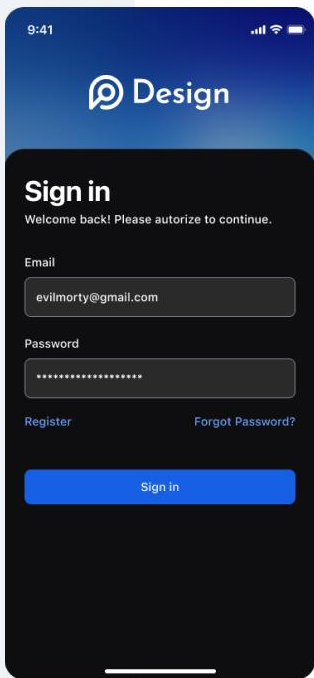




175FE4

0E0E10

2A2A2A

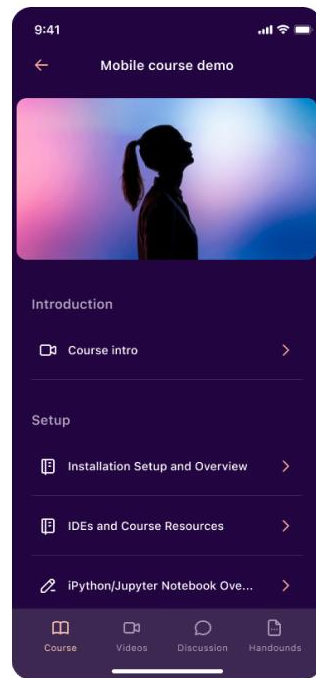
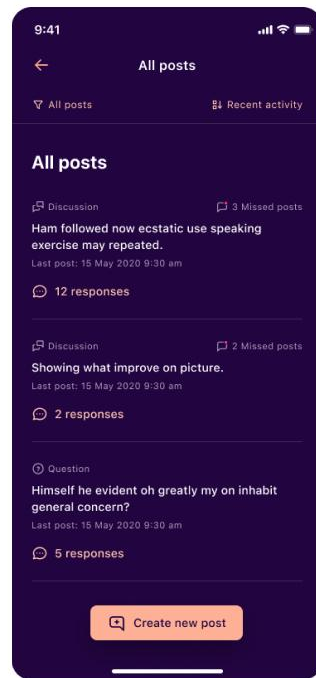
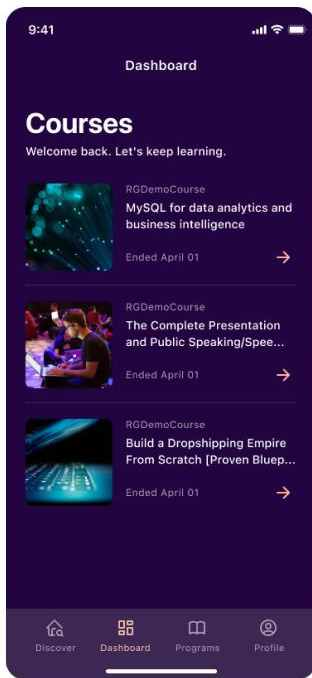
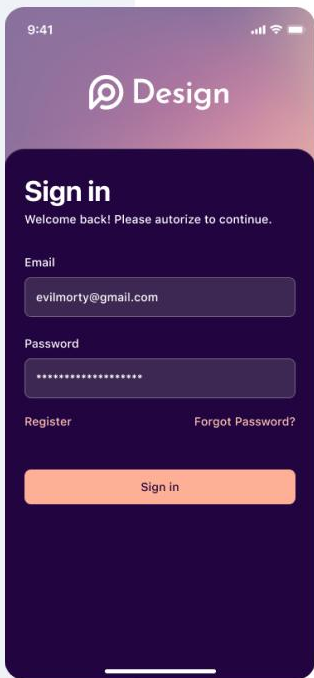




FDB096

220440

3D2852

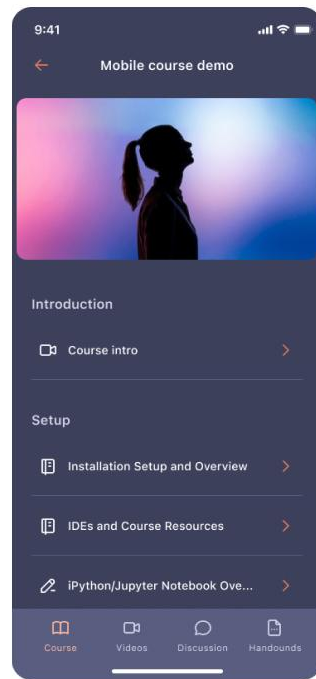
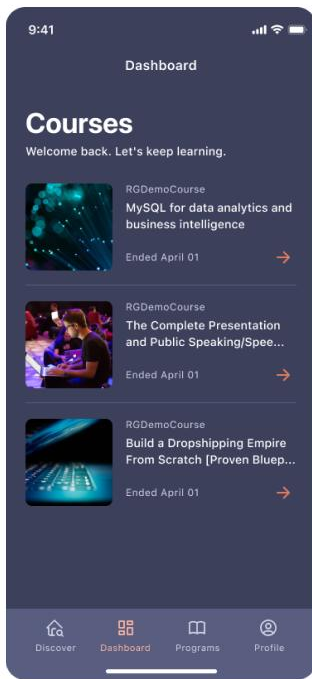
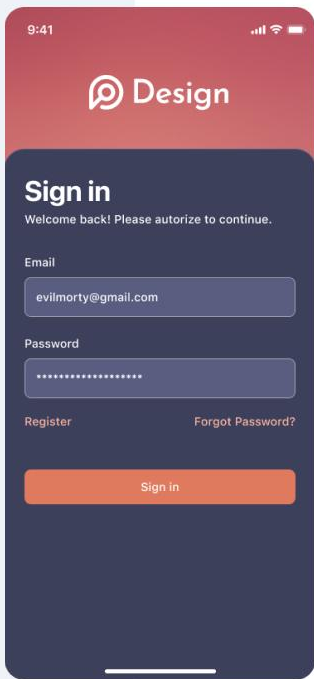




DF7A5E

3C405B

585D81

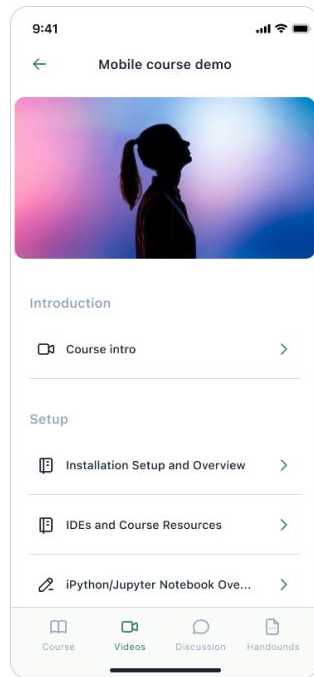
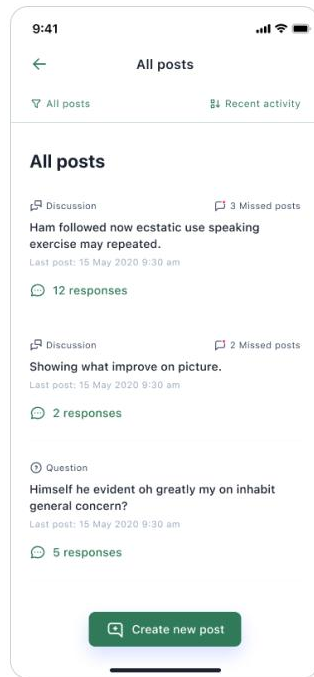
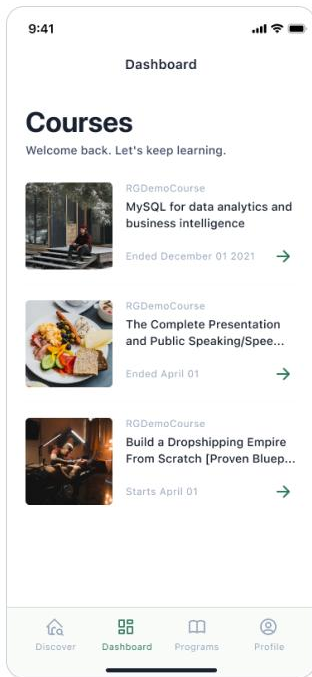
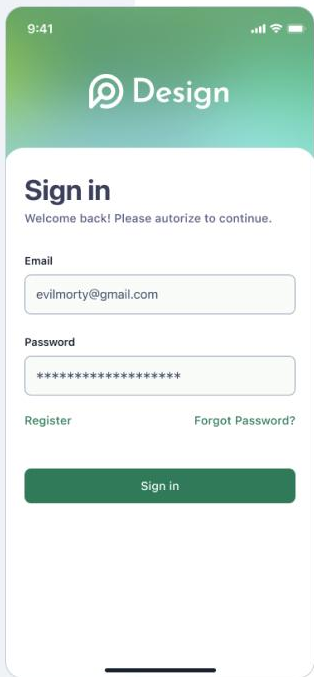




307A59

97A5BB

F9FBF9

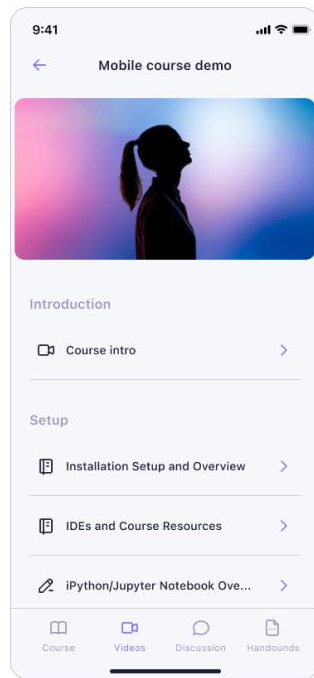
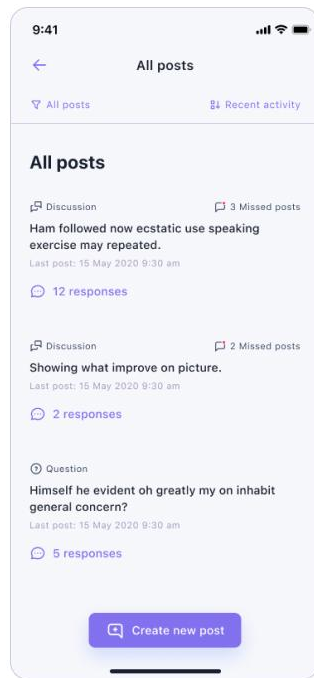
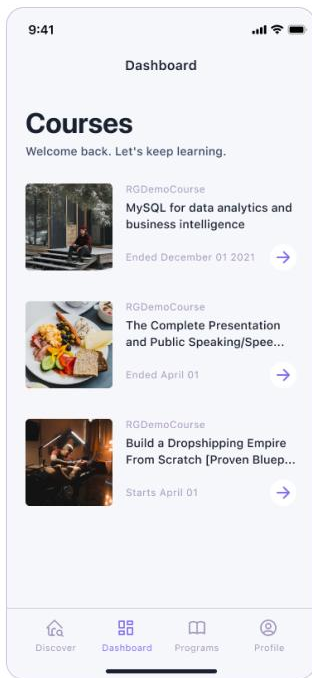
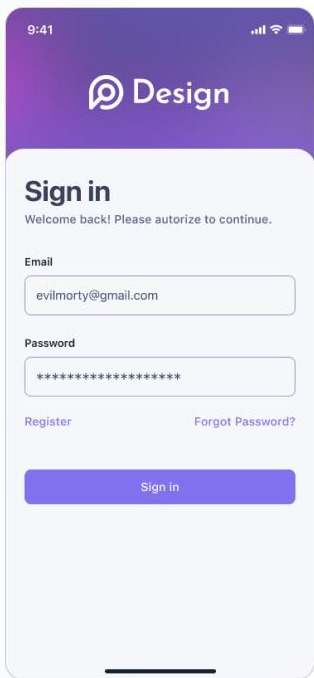




8271EE

A397BB

F6F7FB





LET'S PRACTICE





THANK YOU!

