A REPORT OF SIX WEEKS INDUSTRIAL TRAINING CUM INTERNSHIP

At

ENTERPRISE RESOURCE PLANNING DEPARTMENT

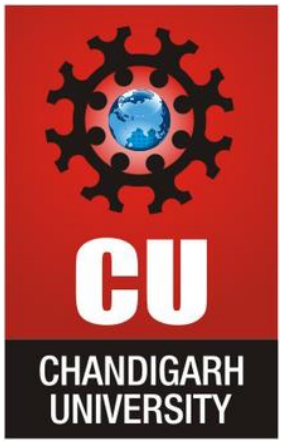
CHANDIGARH UNIVERSITY

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD

OF THE DEGREE OF

BACHELOR OF ENGINEERING

(Computer Science & Engineering)



JUNE-JULY 2018

SUBMITTED BY:

NAME: DEEPAK YADAV

UNIVERSITY UID: 16BCS2082

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY GHARUAN, MOHALI

CANDIDATE'S DECLARATION

I DEEPAK YADAV hereby declare that I have undertaken six weeks institutional training at ERP DEPT CHANDIGARH UNIVERSITY during a period from 2th May 2018 to 17th June 2018 in partial fulfillment of requirements for the award of degree of B.E (COMPUTER SCIENCE & ENGINEERING) at CHANDIGARH UNIVERSITY GHARUAN, MOHALI. The work which is being presented in the training report submitted to Department of Computer Science & Engineering at CHANDIGARH UNIVERSITY GHARUAN, MOHALI is an authentic record of training work.

Signature of the Student

The six weeks industrial training cum Internship Viva-Voice Examination was held on and accepted.

Signature of Internal Examiner Signature of External Examiner

ABSTRACT

Asia's fastest growing university, Chandigarh University (CU) is a leading Indian Institution offering its students a unique amalgamation of professional and academic excellence. Ranked among Asia's best and fastest growing universities, CU has coupled the experience of top industry leaders and renowned academicians and fosters a worldly approach.

Wide spectrum of programs paired with flexibility, experiential learning and interdisciplinary orientation emancipate our students to explore their interests and pursue fulfilling careers. At CU they are grooming students to be socially sensitive through intellectually challenging and contemporary diverse culture.

Chandigarh University admits more than ten thousands students every year in various Degree courses. Admissions in CU is not just a simple task on the university's perspective. The university has divided the admissions into many modules for a simple, easy and effortless admission. As when the admission months are at the summit, it is very important to keep an applicant engaged to the application and tempt him/her in for further processes.

There has been tremendous growth in the use of the mobile phones in youth or let's say, our applicants. Mobile phones are so convenient to use and a needed solution that every teenager has one. With increasing number of features being added to the phone, there has been massive increase in the number of users and hence to new market of Mobile Applications. Android being one of largest market spread has more than 3.3 million applications with more than 33 billion downloads.

As the market is growing, The University needed to have a brand name on the platform of Mobile. And hence the development of "CU FastTrack Admissions, 2019-20" Android Application started.

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my mentors MR. SURINDER DEOL and MR. SHIVENDAR SHARMA who gave me the a special golden opportunity to do this Internship cum Training at Chandigarh university ERP department and helped me building Chandigarh University FastTrack Admissions App.

I would also like to thank my mentor, Prof Sumit Sharma for supporting me for skills development and guidance for android, my parents and friends who helped me a lot in finalizing this project within limited time frame.

ABOUT THE INSTITUTE

CHANDIGARH UNIVERSITY:

Chandigarh University (CU) is a leading Indian Institution offering its students a unique amalgamation of professional and academic excellence. Ranked among Asia's best and fastest growing universities, CU has coupled the experience of top industry leaders and renowned academicians and fosters a worldly approach.

To keep abreast with the global needs and changing market demands, CU have cultivated synergistic alliance with reputed multinational companies to implement an industry-relevant curriculum, based on research and linked to global market outcomes. These strategic partnerships encourage continuous professional learning and provide opportunities for our students to meet and engage with top organizations and industry leaders, to enhance student's educational experience. CU is playing a pro-active role in transforming and constantly elevating the learning process.

ENTERPRISE RESOURCE PLANNING, Chandigarh University:

Enterprise resource planning (ERP) is the integrated management of core business processes, often in real-time and mediated by software and technology.

ERP is usually referred to as a category of business-management software — typically a suite of integrated applications—that an organization can use to collect, store, manage, and interpret data from these many business activities.

ERP provides an integrated and continuously updated view of core business processes using common databases maintained by a database management system. ERP systems track business resources—cash, raw materials, production capacity—and the status of business commitments: orders, purchase orders, and payroll. The applications that make up the system share data across various departments (manufacturing, purchasing, sales, accounting, etc.) that provide the data. ERP facilitates information flow between all business functions and manages connections to outside stakeholders.

Enterprise system software is a multibillion-dollar industry that produces components supporting a variety of business functions. IT investments have become the largest category of capital expenditure in United States-based businesses over the past decade. Though early ERP systems focused on large enterprises, smaller enterprises increasingly use ERP systems.

The ERP system integrates varied organizational systems and facilitates error-free transactions and production, thereby enhancing the organization's efficiency. However, developing an ERP system differs from traditional system development. ERP systems run on a variety of computer hardware and network configurations, typically using a database as an information repository.

LIST OF FIGURES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 About android

Android is a popular computing platform based on the Linux® operating system. The initial commercial version of Android hit the market in 2008 in the form of a mobile phone platform, back when the most popular cell phone for a business user was the BlackBerry, when the iPhone was beginning to make meaningful waves across all sectors, and when the majority of phone users were still tapping out texts from a flip phone.

Android has "paid its dues," so to speak, in the smartphone market for the past decade. The success of Android and iPhone devices has rendered the one-time business mobile device market leader BlackBerry to be the subject of a Bruce Springsteen song: Glory Days. Interestingly, Android's unprecedented success has helped push BlackBerry into a diverse set of offerings, including shipping devices running the Android platform. (Kudos to the BlackBerry team for pivoting and adding value to their shareholders and the broader market despite experiencing the retreat of their earlier dominance.)

In 10 years' time, Android has effectively become the world's most popular operating system by a number of measures. Despite the robust popularity of the flashy and capable Apple iPhone platform, Android shipments worldwide meaningfully outpace Apple's offerings. While Apple's devices continue to demand an ever-increasing price point, Android devices scale the global marketplace. Yes, there are super-pricy Android models sitting next to the latest iPhone, but there are also relatively low-cost Android phones and tablets available for sale at Walmart and on Amazon.

As Android has matured, it is finding its way into a variety of devices, including televisions, projectors, automobiles, and even recreational vehicles. Want to dim the lights in your camper or activate the awning? You can use the Android-based touchscreen interface to manipulate the controls. Or, use your smartphone equipped with Bluetooth to communicate with the RV's Android-based control system. There are many of these types of interfaces finding their way to the market. Some user experiences are simply fantastic (like drone controllers), and some are less than fantastic, like the controls in my uncle’s RV. This article introduces the Android platform and discusses how you can use it for both mobile and non-mobile applications. The ambition is to get

you on a path to making awesome apps for whatever platform arena you feel called to make your contribution.

1.2 Introduction to Activities

The Activity class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a main() method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle.

This document introduces the concept of activities, and then provides some lightweight guidance about how to work with them. For additional information about best practices in architecting your app, see Guide to App Architecture.

The concept of activities

The mobile-app experience differs from its desktop counterpart in that a user's interaction with the app doesn't always begin in the same place. Instead, the user journey often begins non-deterministically. For instance, if you open an email app from your home screen, you might see a list of emails. By contrast, if you are using a social media app that then launches your email app, you might go directly to the email app's screen for composing an email.

The Activity class is designed to facilitate this paradigm. When one app invokes another, the calling app invokes an activity in the other app, rather than the app as an atomic whole. In this way, the activity serves as the entry point for an app's interaction with the user. You implement an activity as a subclass of the Activity class.

An activity provides the window in which the app draws its UI. This window typically fills the screen, but may be smaller than the screen and float on top of other windows. Generally, one activity implements one screen in an app. For instance, one of an app’s activities may implement a Preferences screen, while another activity implements a Select Photo screen.

Most apps contain multiple screens, which means they comprise multiple activities. Typically, one activity in an app is specified as the main activity, which is the first screen to appear when the user launches the app. Each activity can then start another activity in order to perform different actions. For example, the main activity in a simple e-mail app may provide the screen that shows an e-mail

inbox. From there, the main activity might launch other activities that provide screens for tasks like writing e-mails and opening individual e-mails.

Although activities work together to form a cohesive user experience in an app, each activity is only loosely bound to the other activities; there are usually minimal dependencies among the activities in an app. In fact, activities often start up activities belonging to other apps. For example, a browser app might launch the Share activity of a social-media app.

To use activities in your app, you must register information about them in the app's manifest, and you must manage activity lifecycles appropriately. The rest of this document introduces these subjects.

1.3 Intents and Intent Filters

An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases:

Starting an activity

An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an Intent to startActivity(). The Intent describes the activity to start and carries any necessary data.

If you want to receive a result from the activity when it finishes, call startActivityForResult(). Your activity receives the result as a separate Intent object in your activity's onActivityResult() callback. For more information, see the Activities guide.

Starting a service

A Service is a component that performs operations in the background without a user interface. With Android 5.0 (API level 21) and later, you can start a service with JobScheduler. For more information about JobScheduler, see its API-reference documentation.

For versions earlier than Android 5.0 (API level 21), you can start a service by using methods of the Service class. You can start a service to perform a one-time operation (such as downloading a file) by passing an Intent to startService(). The Intent describes the service to start and carries any necessary data.

If the service is designed with a client-server interface, you can bind to the service from another component by passing an Intent to bindService(). For more information, see the Services guide.

Delivering a broadcast

A broadcast is a message that any app can receive. The system delivers various broadcasts for system events, such as when the system boots up or the device starts charging. You can deliver a broadcast to other apps by passing an Intent to sendBroadcast() or sendOrderedBroadcast().

The rest of this page explains how intents work and how to use them. For related information, see Interacting with Other Apps and Sharing Content.

1.4 User Interface & Navigation

Your app's user interface is everything that the user can see and interact with. Android provides a variety of pre-built UI components such as structured layout objects and UI controls that allow you to build the graphical user interface for your app. Android also provides other UI modules for special interfaces such as dialogs, notifications, and menus.

1.5 Animations and Transitions

When your UI changes in response to user action, you should animate the layout transitions. These animations give users feedback on their actions and help keep them oriented to the UI.

Android includes the transitions framework, which enables you to easily animate changes between two view hierarchies. The framework animates the views at runtime by changing some of their property values over time. The framework includes built-in animations for common effects and lets you create custom animations and transition lifecycle callbacks.

1.6 Background processing

Every Android app has a main thread which is in charge of handling UI (including measuring and drawing views), coordinating user interactions, and receiving lifecycle events. If there is too much work happening on this thread, the app appears to hang or slow down, leading to an undesirable user experience. Any long-running computations and operations such as decoding a bitmap, accessing the disk, or performing network requests should be done on a separate background thread. In general, anything that takes more than a few milliseconds should be delegated to a background thread.

When considering doing work off the main thread, consider three factors:

Can the work be deferred, or does it need to happen exactly when scheduled? For example, if you need to fetch some data from the network in response to the user clicking a button, that work must be done right away. However, if you want to upload your logs to the server, that work can be deferred without affecting your app's performance or user expectations.

Once the work starts executing, should the OS try to keep the app process alive? For example, decoding and displaying a bitmap is something that only needs to happen while the app is in the foreground and the process is alive. However, a music player needs to continue playing music even when the app is in the background and not actively being used.

Does the work start in response to system triggers? These triggers could be things like network status, battery state, storage levels, and so on. For example, you might want to wait to communicate with your server until you are out of airplane mode. In this case, if the app process is dead, you might want it to be recreated once the device is online again so that you can perform your work.

1.7 Data and file storage overview

Android provides several options for you to save your app data. The solution you choose depends on your specific needs, such as how much space your data requires, what kind of data you need to store, and whether the data should be private to your app or accessible to other apps and the user.

Internal file storage: Store app-private files on the device file system.

External file storage: Store files on the shared external file system. This is usually for shared user files, such as photos.

Shared preferences: Store private primitive data in key-value pairs.

Databases: Store structured data in a private database.

Except for some types of files on external storage, all these options are intended for app-private data—the data is not naturally accessible to other apps. If you want to share files with other apps, you should use the FileProvider API. To learn more, read Sharing Files.
If you want to expose your app's data to other apps, you can use a ContentProvider. Content providers give you full control of what read/write access is available to other apps, regardless of the storage medium you've chosen for the data (though it's usually a database). For more information, read Content Providers.

1.8 Volley overview

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley is available on GitHub.
Volley offers the following benefits:

Automatic scheduling of network requests.
Multiple concurrent network connections.
Transparent disk and memory response caching with standard HTTP cache coherence.
Support for request prioritization.
Cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel.
Ease of customization, for example, for retry and backoff.
Strong ordering that makes it easy to correctly populate your UI with data fetched asynchronously from the network.
Debugging and tracing tools.

Volley excels at RPC-type operations used to populate a UI, such as fetching a page of search results as structured data. It integrates easily with any protocol and comes out of the box with support for raw strings, images, and JSON. By providing built-in support for the features you need, Volley frees you from writing boilerplate code and allows you to concentrate on the logic that is specific to your app.

Volley is not suitable for large download or streaming operations, since Volley holds all responses in memory during parsing. For large download operations, consider using an alternative like DownloadManager.

1.9 Web-based content

Android offers a variety of ways to present content to a user. To provide a user experience that's consistent with the rest of the platform, it's usually best to build a native app that incorporates framework-provided experiences, such as Android App Links or Search. Additionally, you can use Google Play-based experiences, such as App Actions and Slices, where Google Play services is available. Some apps, however, may need increased control over the UI. In this case, a WebView is a good option for displaying trusted first-party content.

You can provide access to your web pages from either a browser or your own Android app. The WebView framework allows you to specify viewport and style properties that make your web pages appear at the proper size and scale on all screen configurations for all major web browsers. You can even define an interface between your Android app and your web pages that allows JavaScript in the web pages to call upon APIs in your app—providing Android APIs to your web-based application.

However, you shouldn't develop an Android app simply as a means to view your website. Rather, the web pages you embed in your app should be designed especially for that environment.

CHAPTER 2

TRAINING WORK UNDERTAKEN

Abstract:

The project is to build and android application along with web control panels for Admissions in the university.

Software Requirements

Android Studio:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Xampp:

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Sublime Text:

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup

languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

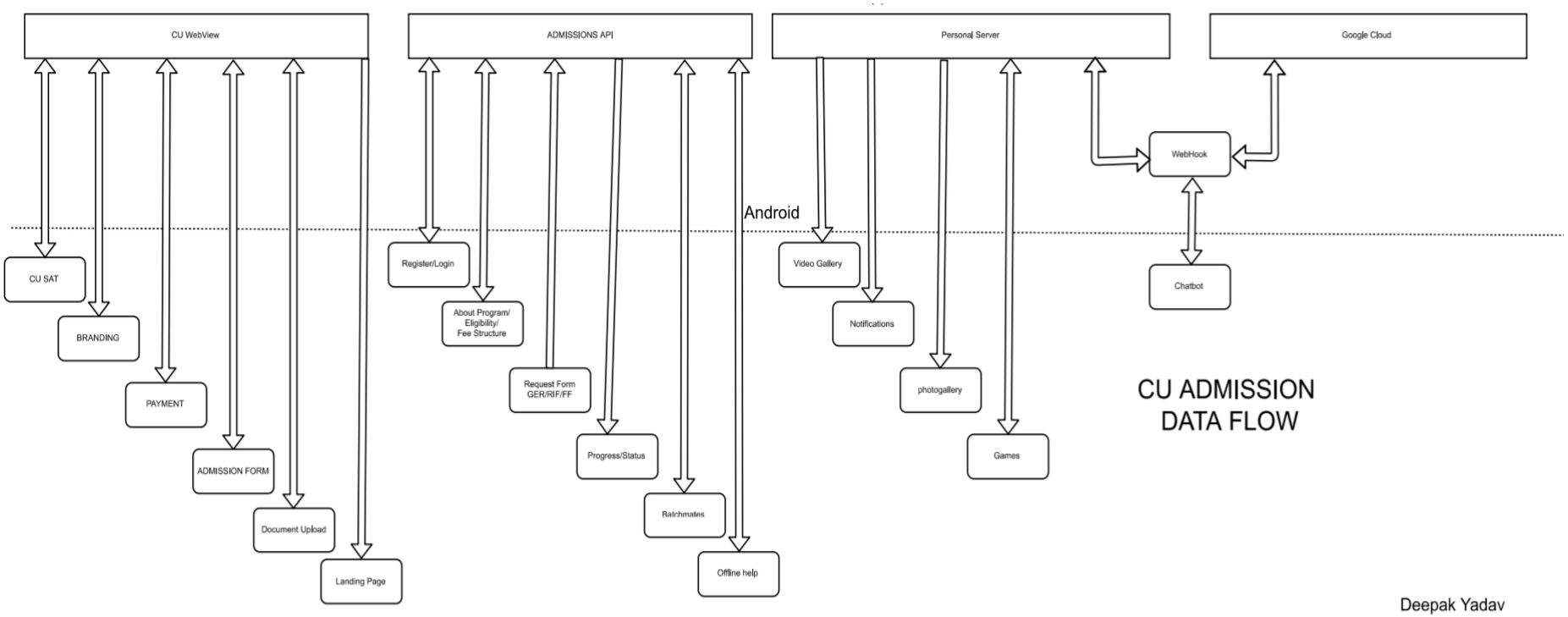
Introduction

Android is a popular computing platform based on the Linux® operating system. The initial commercial version of Android hit the market in 2008 in the form of a mobile phone platform, back when the most popular cell phone for a business user was the BlackBerry, when the iPhone was beginning to make meaningful waves across all sectors, and when the majority of phone users were still tapping out texts from a flip phone.

2.1 SOFTWARE ARCHITECTURE DESIGN

Various Required modules were brain put together and shorted according to the potential of engagement of student to the university during admissions.
The main Application will Run on android powered by web-hooks from University Data APIs.

2.2 USER/DATA FLOW DIAGRAM



Deepak Yadav
16BCS2082
deepakyadav.j23@gmail.com

Fig 1: CU ADMISSIONS DATA FLOW DIAGRAM

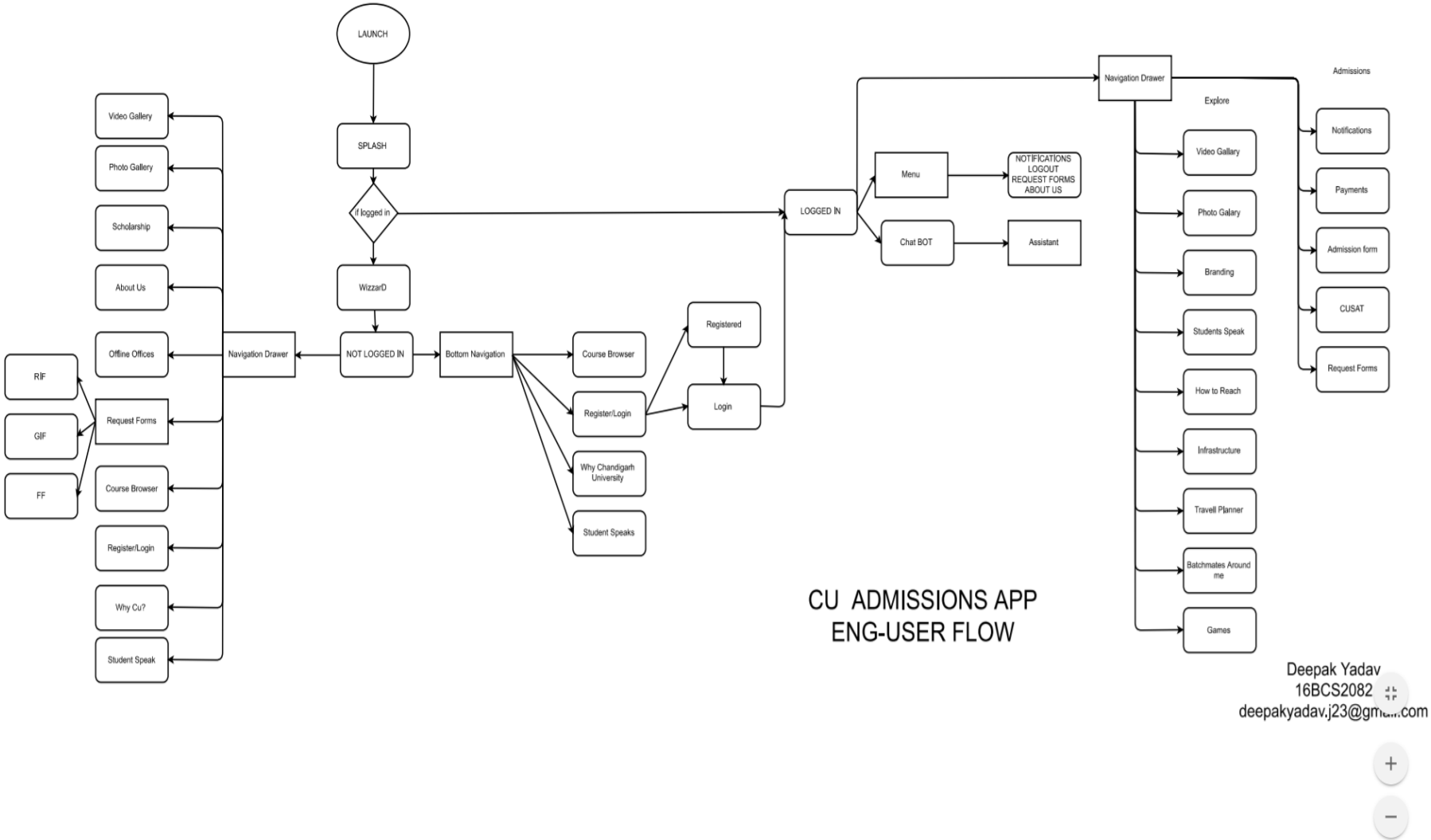


Fig 2: CU ADMISSIONS USER FLOW

2.3 ANDROID

On the android Side, that is the client, various modules have been planned. The Application is divided into two phases, Phase 1 contains all of those modules that a user can use before any registration. Phase 2 contains all the modules that a user can use after he she has been registered for admissions.

Phase 1 modules

- Login
- Register
- Course Structure Browser
- Students Speak
- Why Chandigarh University
- Scholarship Schemes
- Video Gallery
- Photo Gallery
- Admission Offices
- Request Form
- About Us

Phase 2 Modules

- Notifications
- Chat Assistant
- Video gallery
- Photo Gallery
- This is Us Branding Page
- Student Speaks
- How to reach CU
- Travel Planner
- Find a Friend
- Game Box
- Dynamic Menu
- Payments

- Admission form
- Documents Upload
- Request Forms
- About us

2.4 WEB

For adding various elements in the database for photo and video gallery, a Web Interface is created that will be used to upload and video meta/photo meta data and wil be displayed in the client application.

Video gallery

Uploading a video

- YouTube URL
- Title
- Description
- Date of upload
- Category
- Thumbnail

Browser Gallery

Delete/edit Video

Uploading a photo

- Photo
- Title is from the file name
- Date and Category
- Other meta data

Browser Gallery

Delete/edit Photo

CHAPTER 3

RESULTS AND DISCUSSIONS

Chandigarh University FastTrack Admissions App was Successful Built and compiled to many android devices for testing and is working Fine in All the API levels of android.

All the admissions APIs were working fine with caching mechanism by volley.

UI and UX were greatly appreciated and now the application is ready to launch.

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

4.1 CONCLUSION

In this Project, the design and implementation Android APIs are carried out. The main goals is: to optimize the network resources, provide simple clean interface and student attracting interface for android, to provide assistance during admissions, get essential document and forms from student. As a result of this work, the solution implemented can be changed according to current organization requirements. This is especially useful, because the youth is more of a mobile than a Desktop.

4.2 FUTURE SCOPE

Android is Ruling over the market f mobile applications, having an android version of your website is must. In future scope, we can add many more modules to attract the potential students to the university. With help of Artificial Intelligence and NLP we can support many other reginal languages and track where the user is not able to get through.

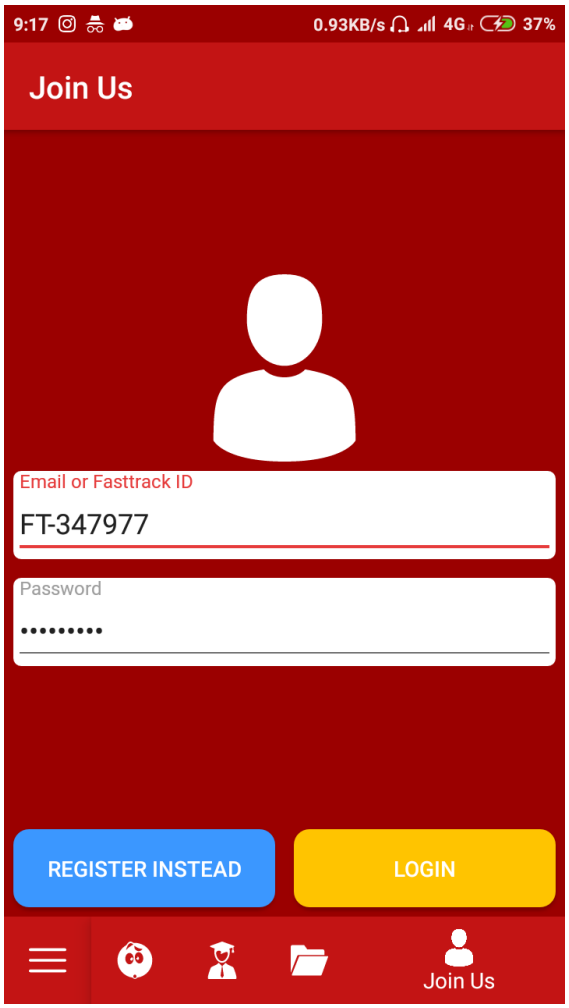
REFERENCES

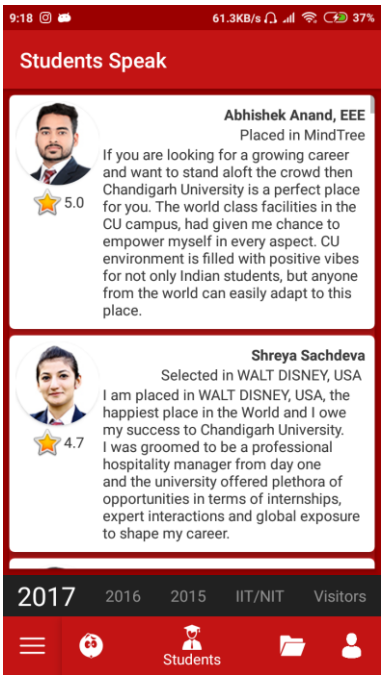
1. Java 10th Edition Complete reference Mc Graw Hill publications

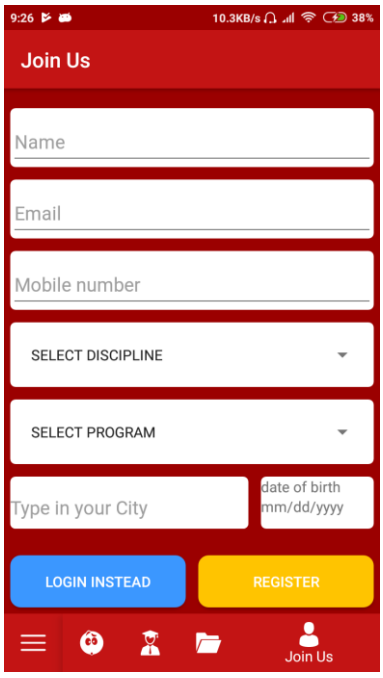
2. https://developer.android.com

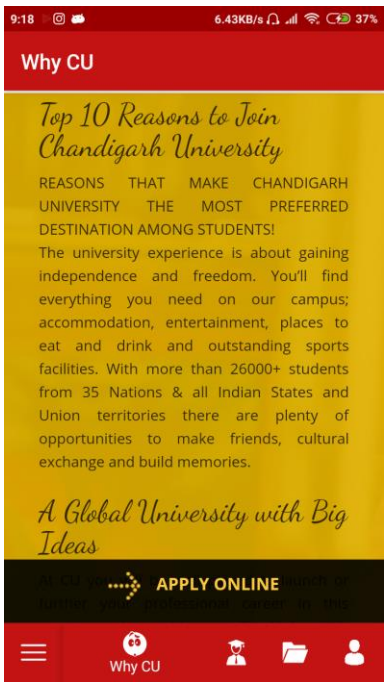
3. https://stackoverflow.com

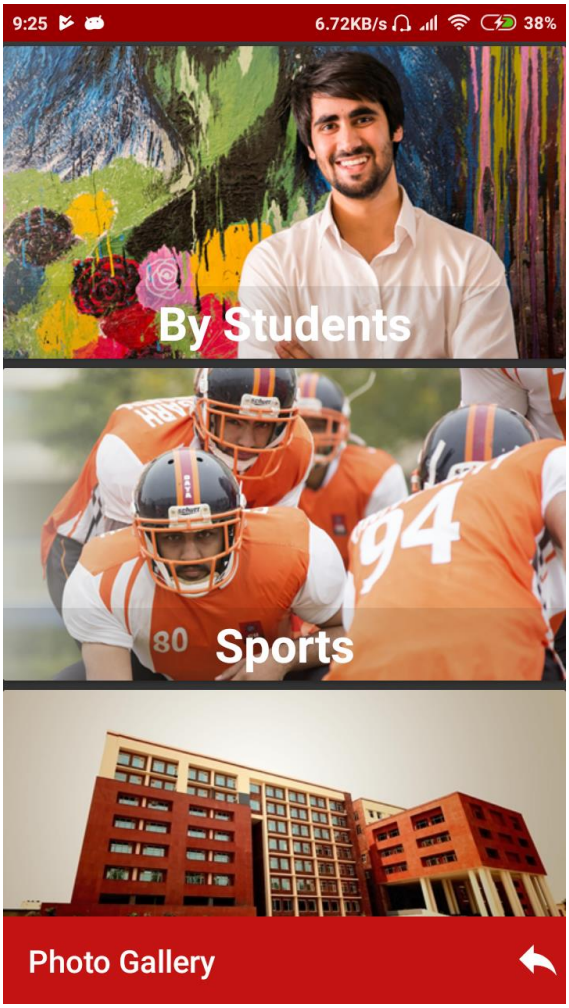
SCREENSHOTS

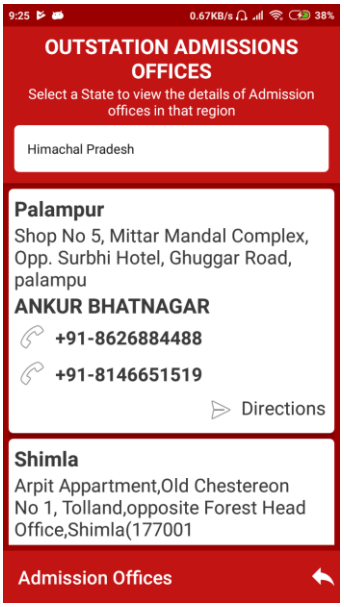




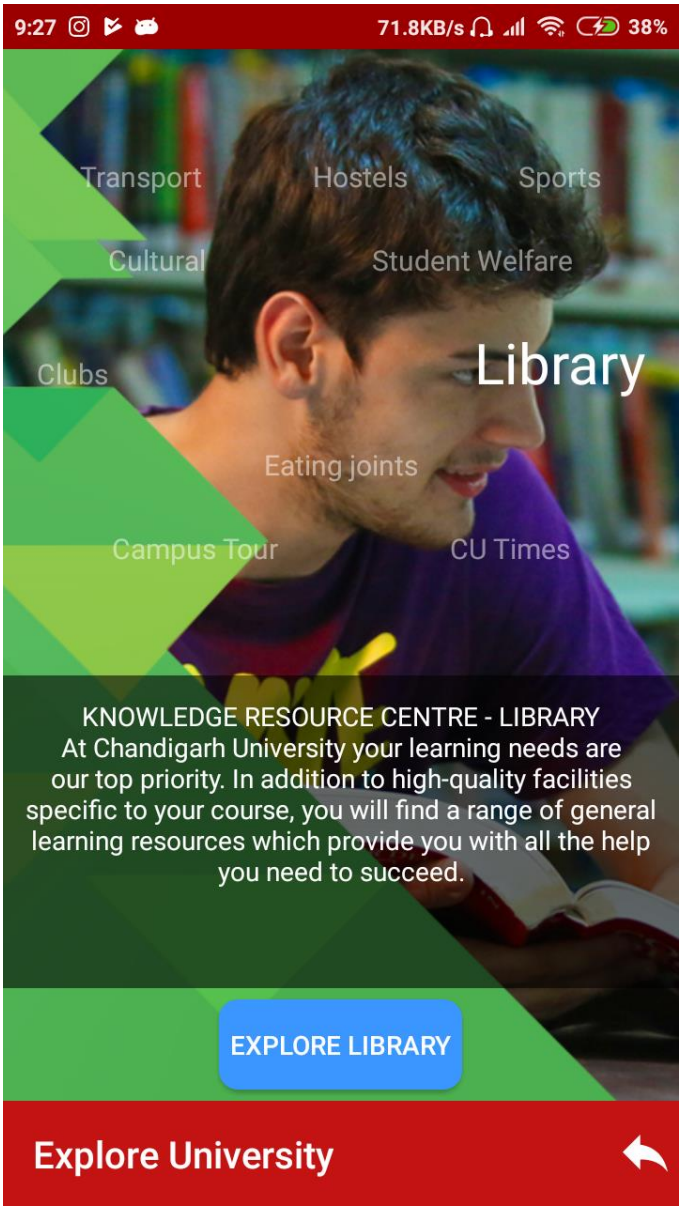


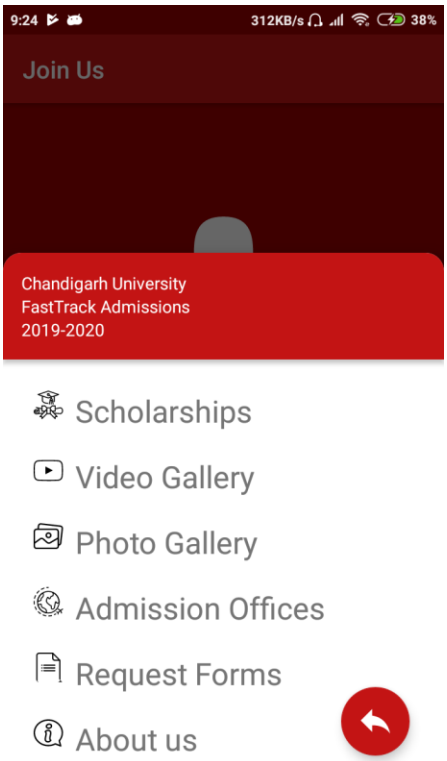


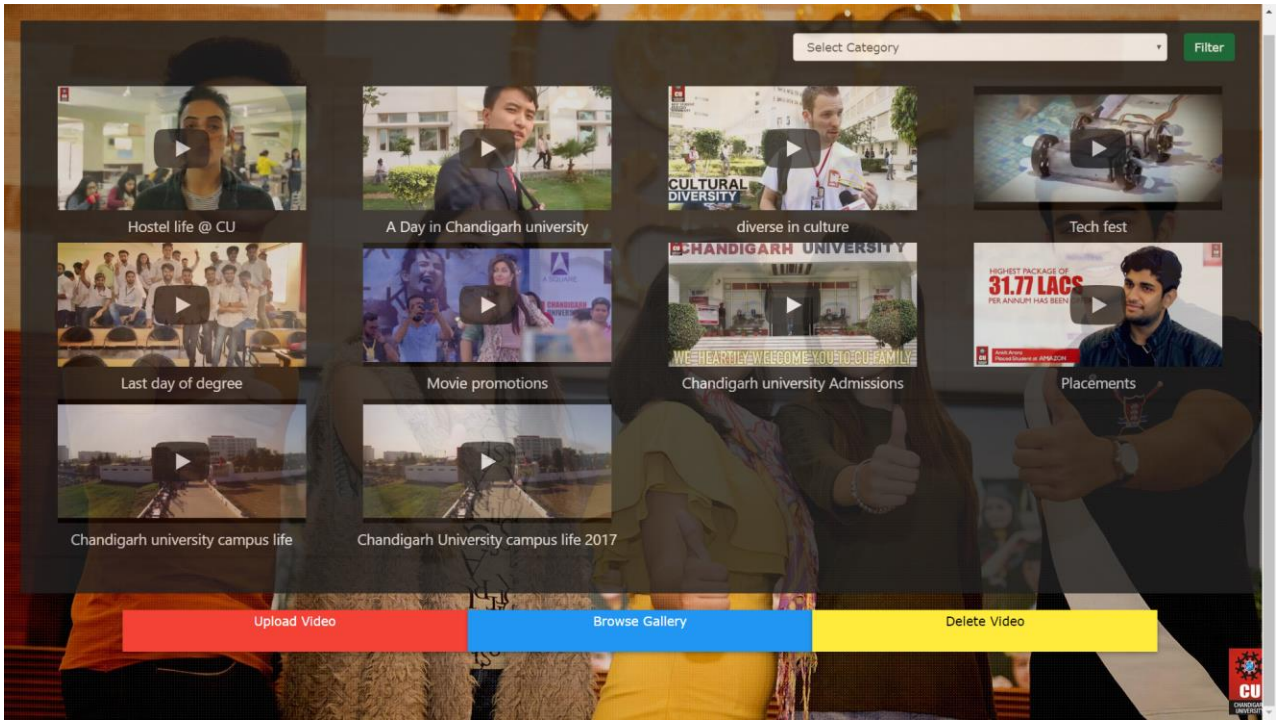

9:25
6.72KB/s
38%
By Students
Sports
Photo Gallery

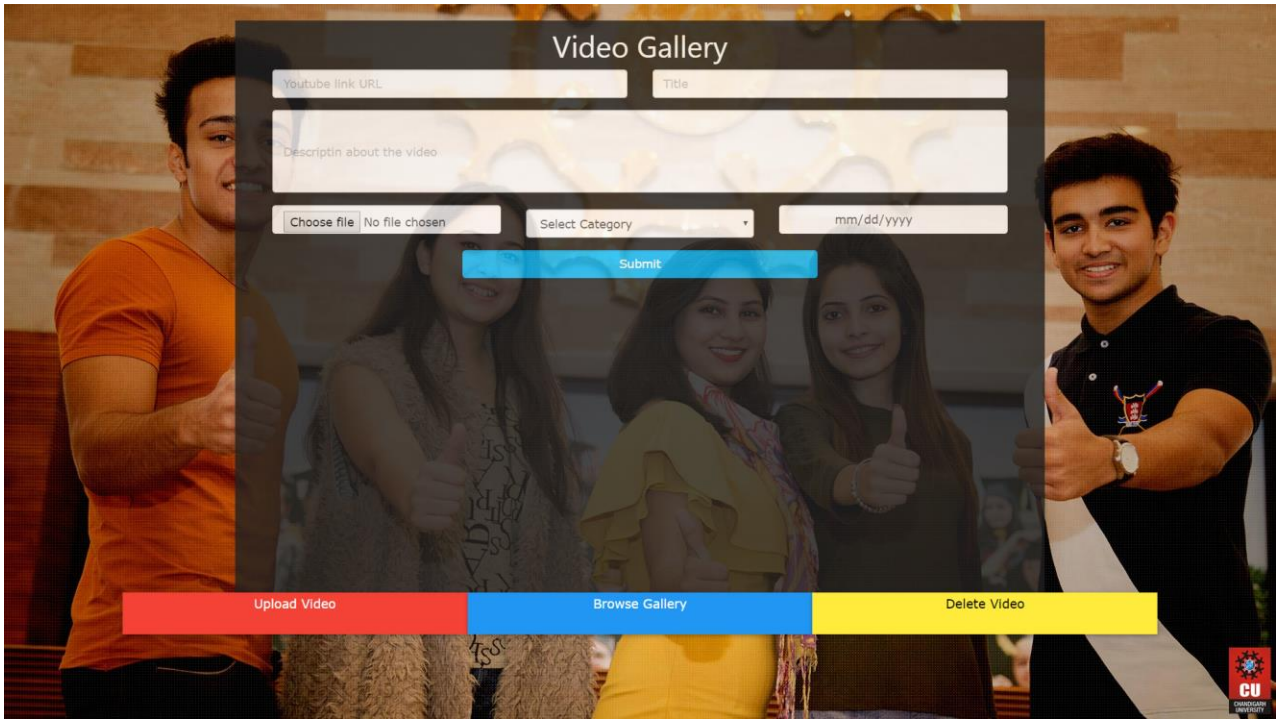

9:25
0.67KB/s
38%
OUTSTATION ADMISSIONS OFFICES
Select a State to view the details of Admission offices in that region
Himachal Pradesh
Palampur
Shop No 5, Mittar Mandal Complex, Opp. Surbhi Hotel, Ghuggar Road, palampu
ANKUR BHATNAGAR
+91-8626884488
+91-8146651519
Directions
Shimla
Arpit Appartment,Old Chestereon No 1, Tolland,opposite Forest Head Office,Shimla(177001
Admission Offices


9:26
0.07KB/s
38%
Get In Touch
Chandigarh University
Email: admissions@cumail.in
Phone: 1800 123 123
NH-95 Gharuan Mohali, Punjab 140413 (India)
General Helpline Number:9876543212
Toll free number: 1800 3124 325
View on Google Maps
About us


9:27
71.8KB/s
38%
Transport
Hostels
Sports
Cultural
Student Welfare
Clubs
Library
Eating joints
Campus Tour
CU Times
KNOWLEDGE RESOURCE CENTRE - LIBRARY
At Chandigarh University your learning needs are our top priority. In addition to high-quality facilities specific to your course, you will find a range of general learning resources which provide you with all the help you need to succeed.
EXPLORE LIBRARY
Explore University


9:24
312KB/s
38%
Join Us
Chandigarh University
FastTrack Admissions
2019-2020
Scholarships
Video Gallery
Photo Gallery
Admission Offices
Request Forms
About us


Select Category
Filter
Hostel life @ CU
A Day in Chandigarh university
diverse in culture
Tech fest
Last day of degree
Movie promotions
Chandigarh university Admissions
Placements
Chandigarh university campus life
Chandigarh University campus life 2017
Upload Video
Browse Gallery
Delete Video


Video Gallery
Youtube link URL
Title
Descriptin about the video
Choose file
No file chosen
Select Category
mm/dd/yyyy
Submit
Upload Video
Browse Gallery
Delete Video