`

A Mini Project Report on

Placement Assistance Service

T.E.-I.T Engineering

Submitted By

GulshanYadav	**20104085**
Gandharvi Walavekar	**20104045**
Vaishnavi Shinde	**20104002**

Under The Guidance Of

Prof. Apeksha Mohite

DEPARTMENT OF INFORMATION TECHNOLOGY

A.P.SHAH INSTITUTE OF TECHNOLOGY
G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic year : 2022-23

`

CERTIFICATE

This to certify that the Mini Project report on **Placement Assistance Service** has been submitted by _Gulshan Yadav_(20104085), Gandharvi Walavekar (20104045) and Vaishnavi Shinde (20104002) who are a Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfilment of the requirement for the degree in **Information Technology**, during the academic year **2022-23**in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Apeksha Mohite
Guide

Dr.Kiran Deshpande	Dr.Uttam D. Kolekar
Head Department of InformationTechnology	Principal

External Examiner(s)
1.
2.

Place: A.P.Shah Institute of Technology, Thane
Date:

`

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Prof. Apeksha Mohite** Expressing gratitude towards our HoD, Dr**. Kiran Deshpande**, and the Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our teacher Ms. **Yaminee Patil** who gave us her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

`

ABSTRACT

In this competitive era, the education among the people is so increasing that the jobs for them are now decreasing. The companies even want the people who are best in their fields. At that time, it becomes difficult to find the people who are intelligent enough to be hired. The work for the companies also increases to find the people who can fulfill their requirements. Thinking about these problems, one can think about the process which can handle this process and make the work less complex.

A placement assistant service is a tech solution – that assists users in improving their overall procedure. Therefore it addresses aspects related to job activity.As users became more conscious over time,placement assistant service have come to manage a greater number of features.

`

TABLE OF CONTENTS

1. Introduction... 1

    1.1. Purpose...1

    1.2. Problem Statement... 2

    1.3. Objectives.. 2

    1.4. Scope...3

2. Literature Review.. 4

3. Proposed System.. 7

    3.1. Features and Functionality...7

4. Requirements Analysis.. 9

5. Project Design.. 10

    5.1. Use Case diagram... 10

    5.2. DFD (Data Flow Diagram).. 11

    5.3. System Architecture..12

6. Technical specification.. 13

7. Project Scheduling.. 15

8. Implementation... 16

9. Result and Discussion... 17

10. Conclusion and Future Scope.. 18

References

`

Chapter No: 1

INTRODUCTION

Placement Assistance Service (PAS) as the name suggests, the primary focal point is placed in assistance.The main aim of this project is to facilitate the placement procedure in our college. This project reduces manual work and maximizes the optimization, abstraction and security. This is a web application which helps employee as well as the employer authority to carry out the process with ease.

An assistance service is a website dedicated for online information about recruiters as well as job seekers. A service helps both the job seekers and recruiters finding the right organization for the employees. In the case of job seekers, according to their educational qualification, experience and their preferences, the job portal shows the list of companies to the job seeker. And, to the recruiters, provides the suitable candidates from a pool of lacks. The objective of this application is to develop a system to enable interaction between employers and applicants. The determination is to allow communication between the interested parties and complete the task of recruitment quickly.

In this time of recession where everyone, is either experienced or fresher, is in search for a job. This service can prove to be very helpful since it allows users of different profile to search for jobs on the basis of their qualification. Every user can access and apply. Now-a-days the job market is so extensive that a variety of industries and companies are searching for right candidates and the prospective candidates are searching for right companies for growth opportunities. This purpose is served by most of the assistance services on line.

Finding jobs that best suits the interests and skill set is quite a challenging task for the job seekers.The difficulties arise from not having proper knowledge on the organization’s objective, their work culture and current job openings. In addition, finding the right candidate with desired qualifications to fill their current job openings is an important task for the recruiters of any organization. Placement Assistance Services have certainly made job seeking convenient on both sides. The Service is the solution where recruiter as well as the job seeker meet aiming at fulfilling their individual requirement. They are the cheapest as well as the fastest source of communication reaching wide range of audience on just a single click irrespective of their geographical distance.

Purpose:

The purpose of developing an Placement Assistance Service comes from the idea to make the job search efficient and handy. It helps the recruiters as a primary source of talent search. It also helps the job seekers to search for current vacancies at a single point. Therefore, we can say that Placement Assistance Service acts as a bridge of communication between organizations and applicants. With the evolution of technology and internet being the main source of information for the applicants, these services have become an excellent method to reach wide range of audience. Initially, when we were unaware of these services, we used to do research about companies and their technology stack through their respective websites and apply if the job responsibilities matches the interests. This requires lots of effort and time. However, later when we realized the importance of assistance services,we are able to access jobs in companies, locations that we might not otherwise have learned.

Problem Definition:

Some of the common problems faces in Placement Assistance Service are as follows:

- Many of the jobs are not real i.e., they are fake which are listed in the portals.
- The portals do not take responsibility of the jobs listed on the portals whether they are genuine or not.
- Users have to travel to the place where industry is located several times before the final recruitment which wastes their time and money.
- Some of the job portals ask for the money before the commencement of the job.
- The companies listed do not give their actual structure and environment where the industry is located.
- Delay in getting a revert back from the company.

`

OBJECTIVES

- To improve the user’s experience.
- To browse jobs according to personal preference.
- To reduce time usage.
- To make information available at just one click.
- To make it feasible to post jobs.
- To create a user friendly experience.

`

SCOPE

- The service exists to provide flexibility to the jobs seekers by providing the functionalities of both job search and job application in a single application.
- In addition, this application provides an effective means for the employers to post job vacancies and view the job applications by the interested applicants in a single application**.**
- This service is a sound idea or approach for strengthening or retaining relationships with the applicants while reducing time.
- An assistance service is important not just for dealing with existing applicants, but also for obtaining new ones.

Chapter No: 2

Literature Review

The research work done by Marjan and Noziran(pg,2). A study done in 2006, found that 21% of internet users in the EU used the web to search for jobs or to send job applications. In 2007, this had increased to 67% for unemployed people [5]

Mochol and Nixon [9] stated that the use of semantic web technology gives market transparency, higher speed of procurement but reduced transaction cost. Today, the Internet is used for a large number of business transactions. People find the Internet to be an effective communication tool.

In a report in 2005, it was found that 90% of job seekers in Germany use the internet to look for jobs.

Galanaki [5] lists the following methods to be the traditional (old) ways for recruitment:

- Employment recruitment agencies
- Job fairs
- Advertising in the mass media such as newspapers
- Advertisement in television and radio
- Management Consultants
- Existing employee contacts
- Schools colleges or universities students services department
- Workers or professional referrals

These old job seeking methods are too slow, stressful,challenging and also lack quality [6].

Gangle [4] defined the concept of online recruitment or e-recruitment as the use of the Internet to search for jobs which have been advertised electronically. Thus, the employers advertise the job opportunities, save the resume and curriculum vitae (CV) of applicants, contact the jobseekers who are qualified, online.

`

Chapter No: 3

PROPOSED SYSTEM

Features & Functionality:

- Providing Interactive Dashboard

  PAS dashboards are interactive, giving each user the opportunity to view and browse through a collection of jobs posted on the shared platform. A well-designed PAS dashboard provides a snapshot of the jobs posted related to the user's personal preference.

- Option Of Searching/Browsing for Jobs

  Searching for the job based on your own personal preference is made easier.

- Option For Posting a Job

  With job postings, you can display a job according to the company requirements with just one single click.

`

Chapter No: 4

REQUIREMENT ANALYSIS

Importance of Requirements Gathering:

Requirements gathering is a fundamental part of any business decision. It helps generate a list of system, functional and technical requirements from the different stakeholders involved in the process. Being confident about what requirements to look for ensures your expectations with the deliveries are clear, and that eventually enables you to make the right choice when it comes to selecting a solution for your business. No one knows your business better than us. So, it's important to figure out your expectations from the platform before you start looking for one the requirements. Once we know what we want the software to do for our project, so it becomes way easier to pick solutions that line up with your needs.

Need Analysis:

Before we begin, it's important to lay the proper foundation. Knowing which requirements are most important for our company will make us search more effective and less time-consuming.

What goals are you trying to accomplish?

Are you looking for a full-fledged system, or just enough to replace your spreadsheet or pen-and-paper approach?

What size is your project?

Not all systems are built equal. Some cater to small tasks while others are much more suitable for large enterprises. Portals offer a wide range of functionality, and with that comes varying levels of complexity.

Key Requirements:

Through extensive experience and research, we've compiled a comprehensive evaluation checklist of the most critical requirements to look for when researching and comparing tasks for the portal. These requirements are most common for the average website, but some may not be applicable to everyone. Different systems focus on different capabilities, so it's vital to know what functionality your organization.

The following are a checklist of typical CRM requirements in our System:

- Database Management
- Security

`

- Deploy Environment (WebBased)

Functional Requirements:

It is the primary requirement that is fulfilled by our job portal. It's allowing the users, and employers to use our portal at the level ease. The purpose of our portal is to provide the full information that is required of the user. Here is the following requirement that is fulfilled by our system**.**

- **User Login:**

This feature is used by the user/admin to login into the system. A user/admin must log in with his user name and password to the system after registration. If they are invalid, the user is not allowed to enter the system.

- **Register New User:**

A new user will have to register in the system by providing essential details in order to apply in the job.

- **Search Jobs:**

The user can search for the desired jobs. They can view different jobs. After confirming the login in the search user can select it and apply by providing the necessary details.

- **Admin**
- **Post Jobs:**

The administrator can add jobs.

- **ManageDetails:**

The system must identify the login of the admin. Admin account should be secured so that only the owner of the portal can access that account.

Projectrequirementsgatheringisn'taone-and-doneactivity.As your portal evolves and different needs emerge you will require new functionality. It's easy to feel overwhelmedbythedifferentneedsandrequestsbutwithcarefulmanagement,you can evolve your portal while keeping team members happy.

`

Chapter No: 5

PROJECT DESIGN

- **Use Case Diagram**

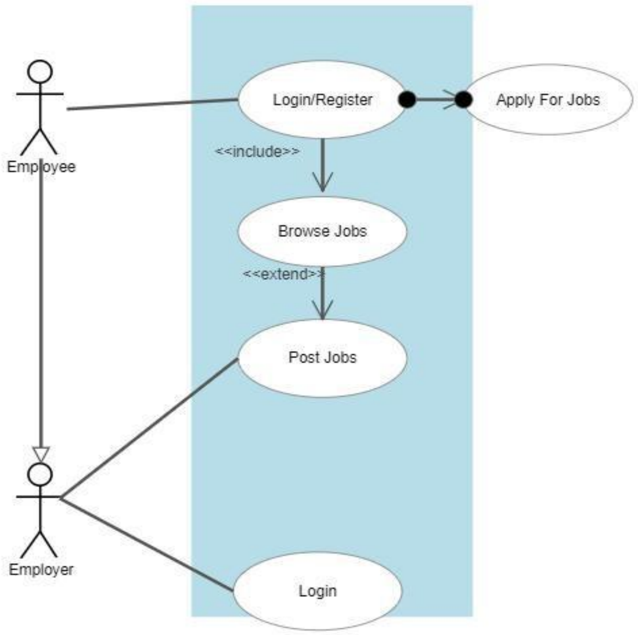


Figure 1: Use Case Diagram

- DFD (Data Flow Diagram)

`

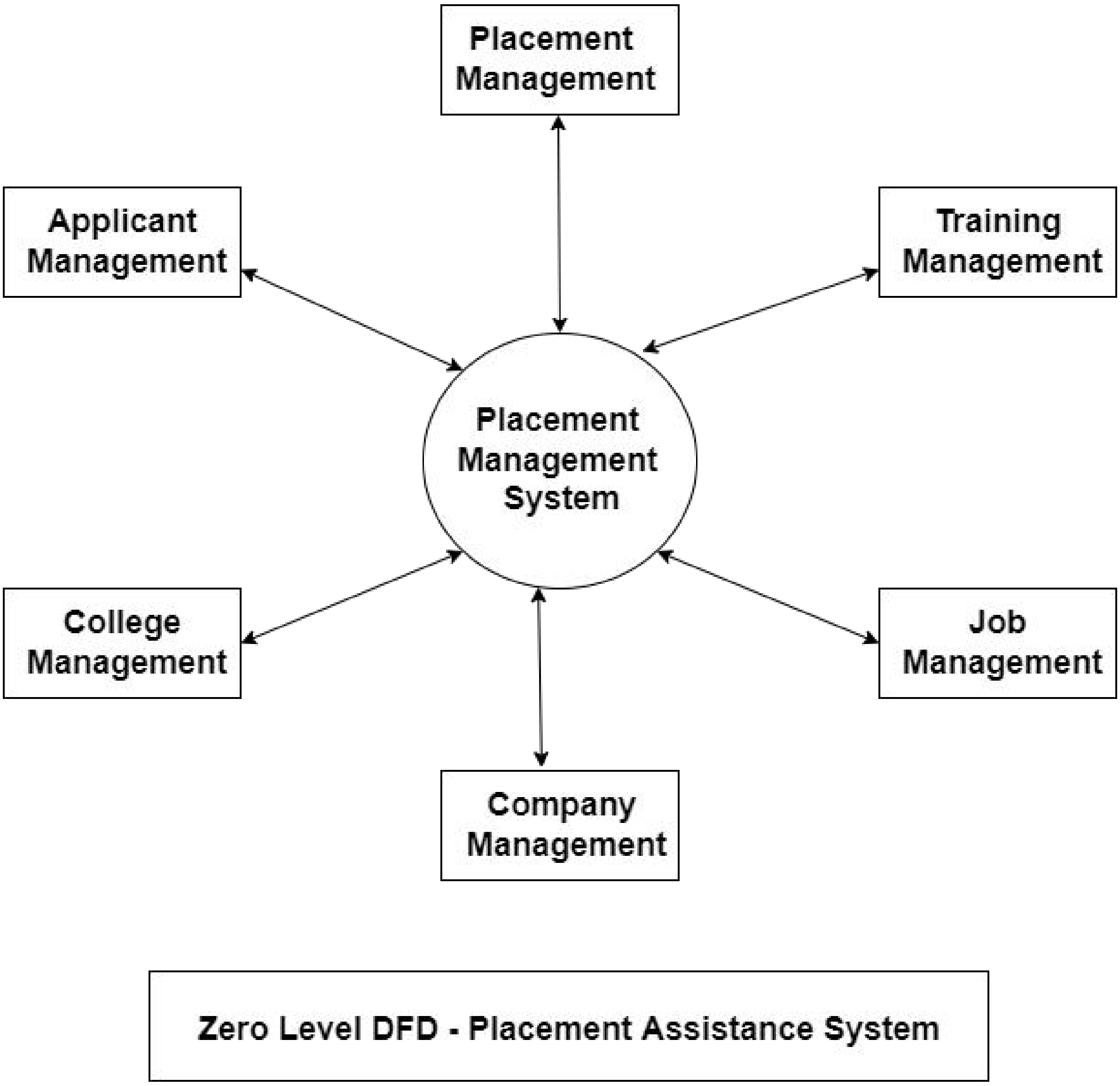


Figure 2: DFD (Level 0)

`

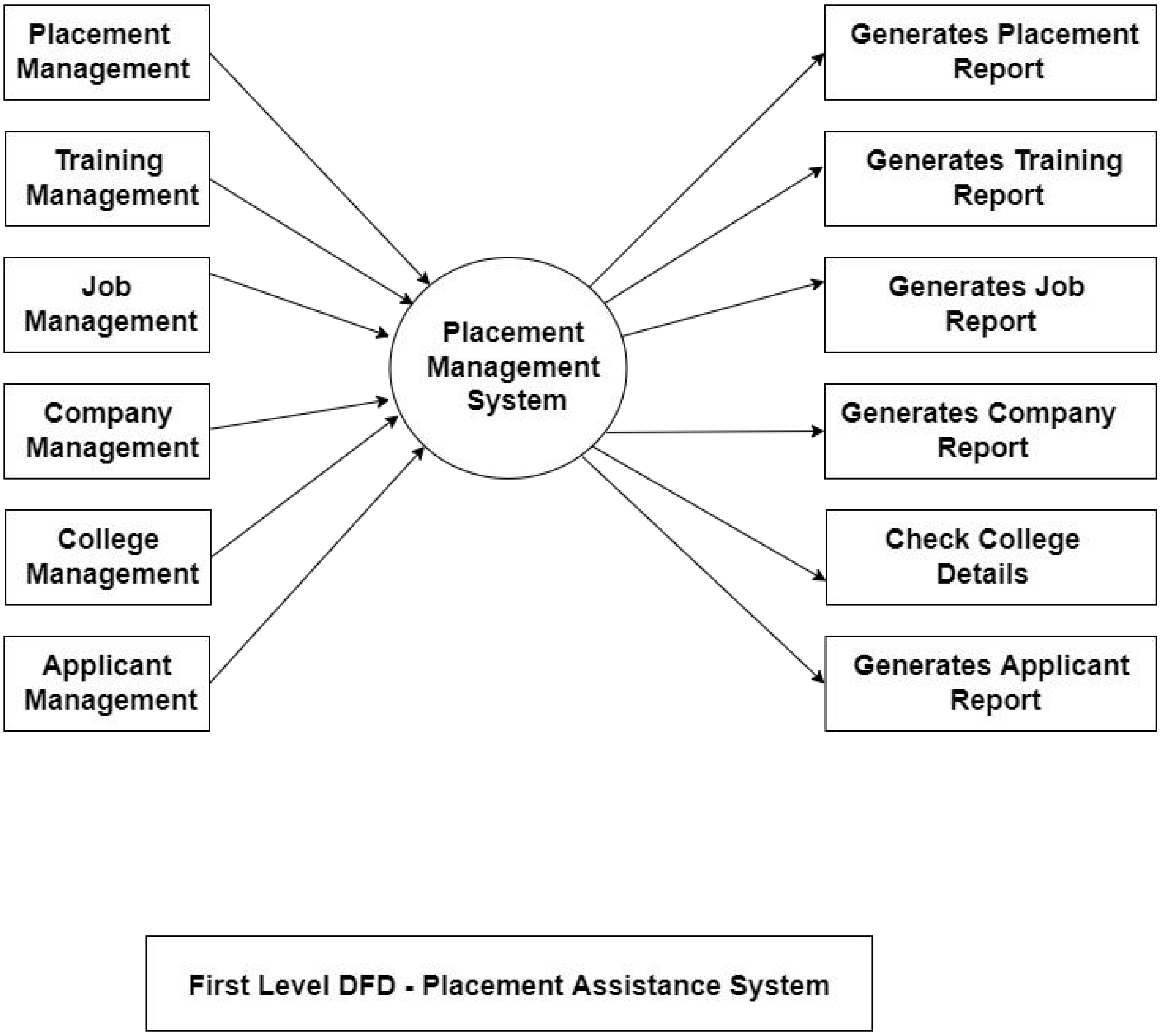


Figure 3: DFD (Level 1)

`

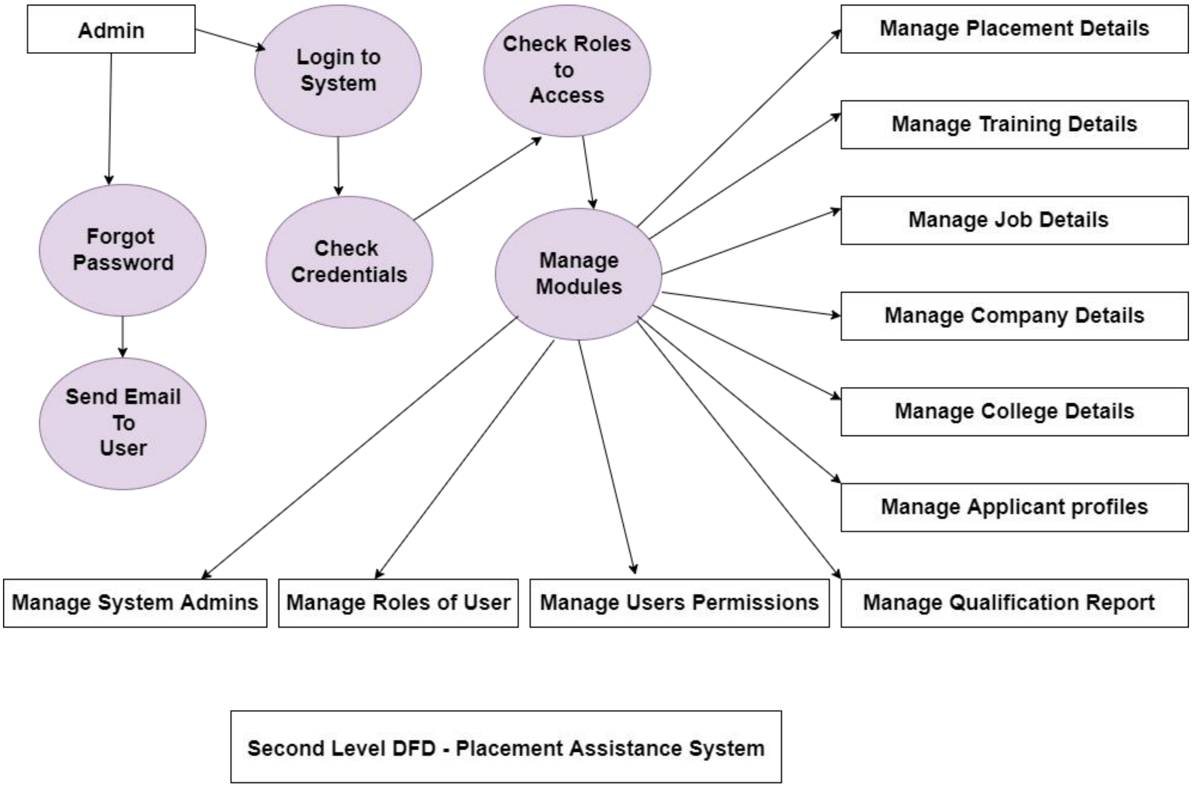


Figure 4: DFD (Level 2)

`

Chapter No.6:

TECHNICAL REQUIREMENTS

Development: VS Code

VS Code also known as Visual Studio Code is a source code editor made by Microsoft for Windows, Linux, MacOS. It has various features such as Debugging, Syntax highlighting, extension, intelligent code completion.

Flutter: We have used flutter since it is an open source framework for making mobile applications.

Backend: Golang

Go (also called Golang or Go language) is an open source programming language used for general purpose. We have used it to create dependable and efficient software.

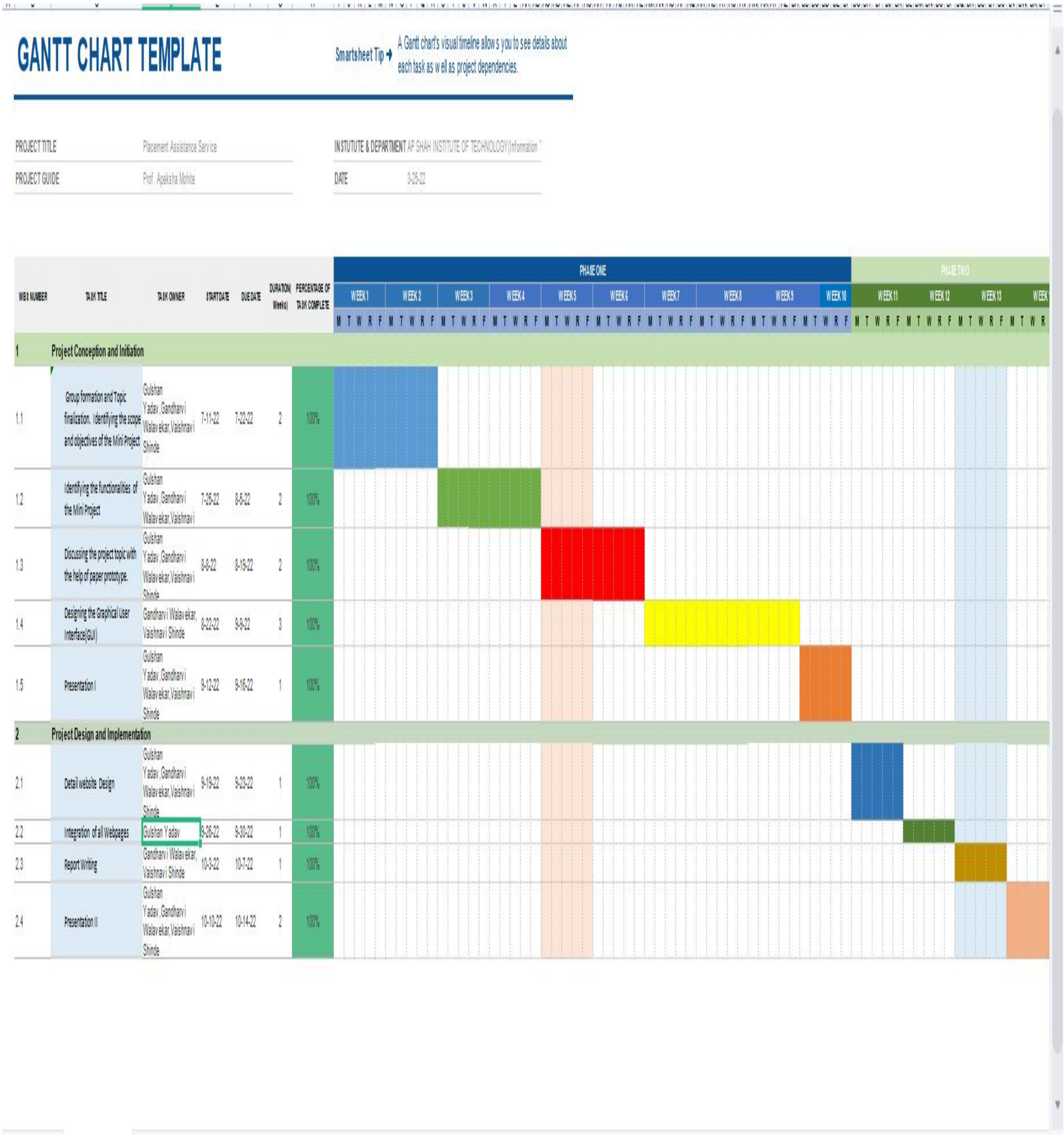
Database: Postgresql,

We have used PostgreSQL since it is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications.

`

Chapter No:7

PROJECT SCHEDULING



GANTT CHART TEMPLATE

Smartsheet Tip ➜ A Gantt chart's visual timeline allow s you to see details about each task as w ell as project dependencies.

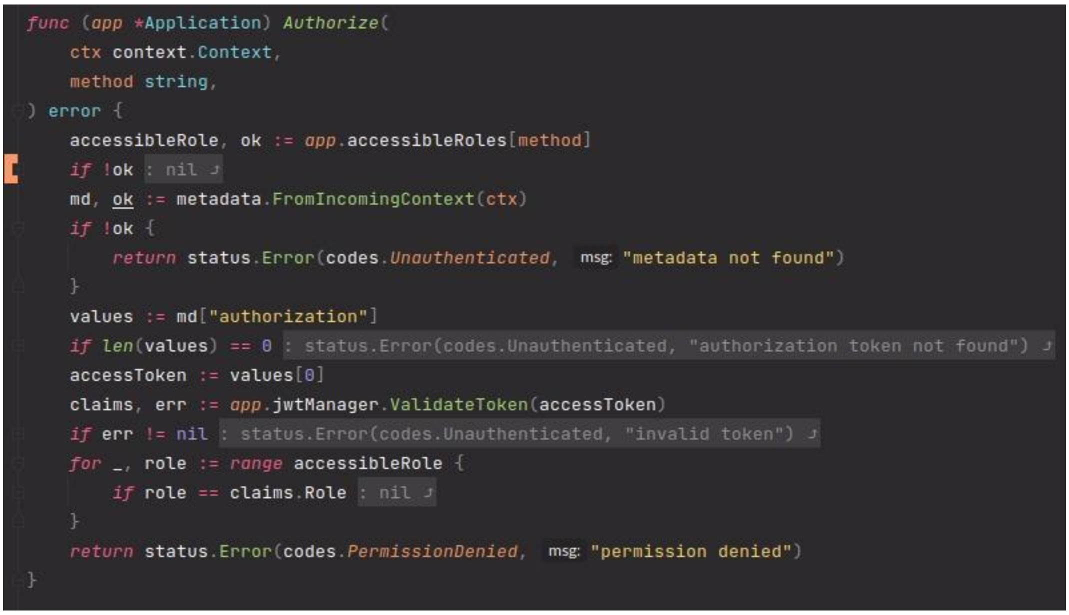
PROJECT TITLE	Placement Assistance Service	INSTUTUTE & DEPARTMENT	AP SHAH INSTITUTE OF TECHNOLOGY(Information
PROJECT GUIDE	Prof. Apeksha Mohite	DATE	3-25-22

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION(Weeks)	PERCENTAGE OF TASK COMPLETE
1	Project Conception and Initiation					
1.1	Group formation and Topic finalization. Identifying the scope and objectives of the Mini Project	Gulshan Yadav ,Gandharvi Walavekar,Vaishnavi Shinde	7-11-22	7-22-22	2	100%
1.2	Identifying the functionalities of the Mini Project	Gulshan Yadav ,Gandharvi Walavekar,Vaishnavi	7-25-22	8-5-22	2	100%
1.3	Discussing the project topic with the help of paper prototype.	Gulshan Yadav ,Gandharvi Walavekar,Vaishnavi Shinde	8-8-22	8-19-22	2	100%
1.4	Designing the Graphical User Interface(GUI)	Gandharvi Walavekar, Vaishnavi Shinde	8-22-22	9-9-22	3	100%
1.5	Presentation I	Gulshan Yadav ,Gandharvi Walavekar,Vaishnavi Shinde	9-12-22	9-16-22	1	100%
2	Project Design and Implementation					
2.1	Detail website Design	Gulshan Yadav ,Gandharvi Walavekar,Vaishnavi Shinde	9-19-22	9-23-22	1	100%
2.2	Integration of all Webpages	Gulshan Yadav	9-26-22	9-30-22	1	100%
2.3	Report Writing	Gandharvi Walavekar, Vaishnavi Shinde	10-3-22	10-7-22	1	100%
2.4	Presentation II	Gulshan Yadav ,Gandharvi Walavekar,Vaishnavi Shinde	10-10-22	10-14-22	2	100%

`

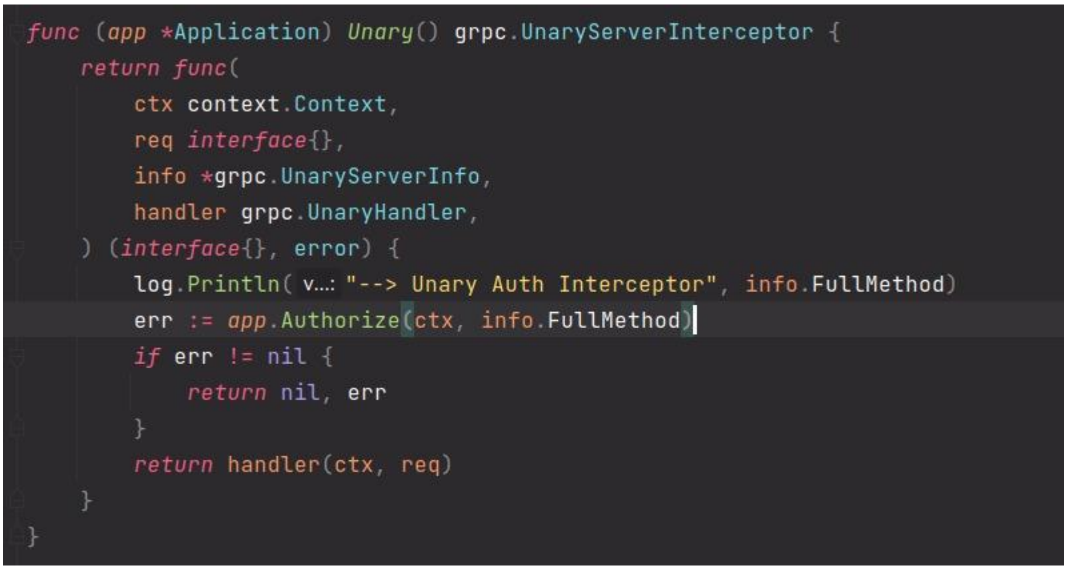
Chapter No:8

IMPLEMENTATION



```
func (app *Application) Authorize(
    ctx context.Context,
    method string,
) error {
    accessibleRole, ok := app.accessibleRoles[method]
    if !ok : nil ⤶
    md, ok := metadata.FromIncomingContext(ctx)
    if !ok {
        return status.Error(codes.Unauthenticated, msg: "metadata not found")
    }
    values := md["authorization"]
    if len(values) == 0 : status.Error(codes.Unauthenticated, "authorization token not found") ⤶
    accessToken := values[0]
    claims, err := app.jwtManager.ValidateToken(accessToken)
    if err != nil : status.Error(codes.Unauthenticated, "invalid token") ⤶
    for _, role := range accessibleRole {
        if role == claims.Role : nil ⤶
    }
    return status.Error(codes.PermissionDenied, msg: "permission denied")
}
```

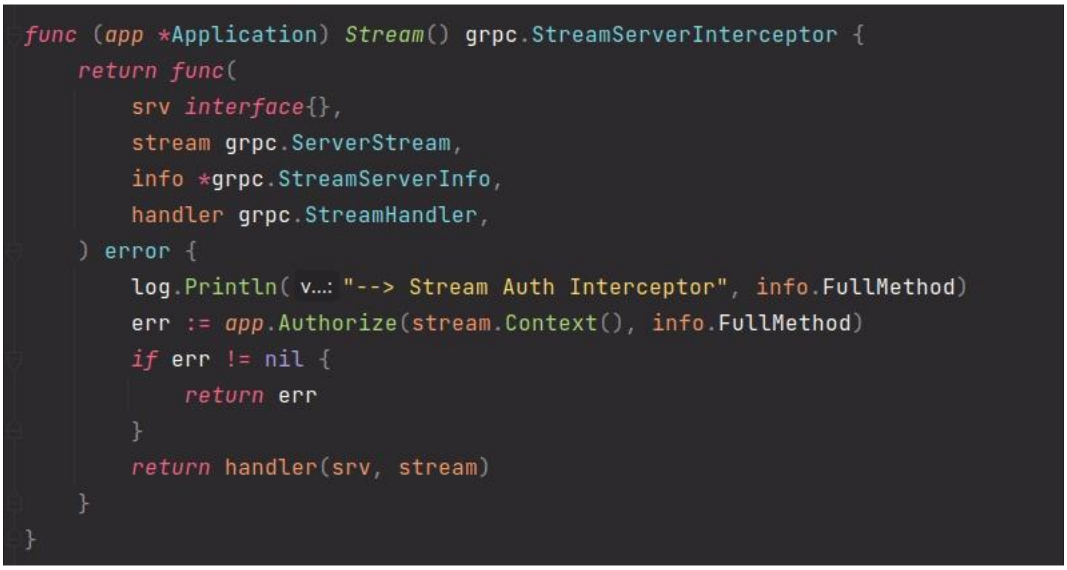
1. This function authorizes the request if they're allowed to access certain service or not based on their role.



```
func (app *Application) Unary() grpc.UnaryServerInterceptor {
    return func(
        ctx context.Context,
        req interface{},
        info *grpc.UnaryServerInfo,
        handler grpc.UnaryHandler,
    ) (interface{}, error) {
        log.Println( v...: "--> Unary Auth Interceptor", info.FullMethod)
        err := app.Authorize(ctx, info.FullMethod)
        if err != nil {
            return nil, err
        }
        return handler(ctx, req)
    }
}
```

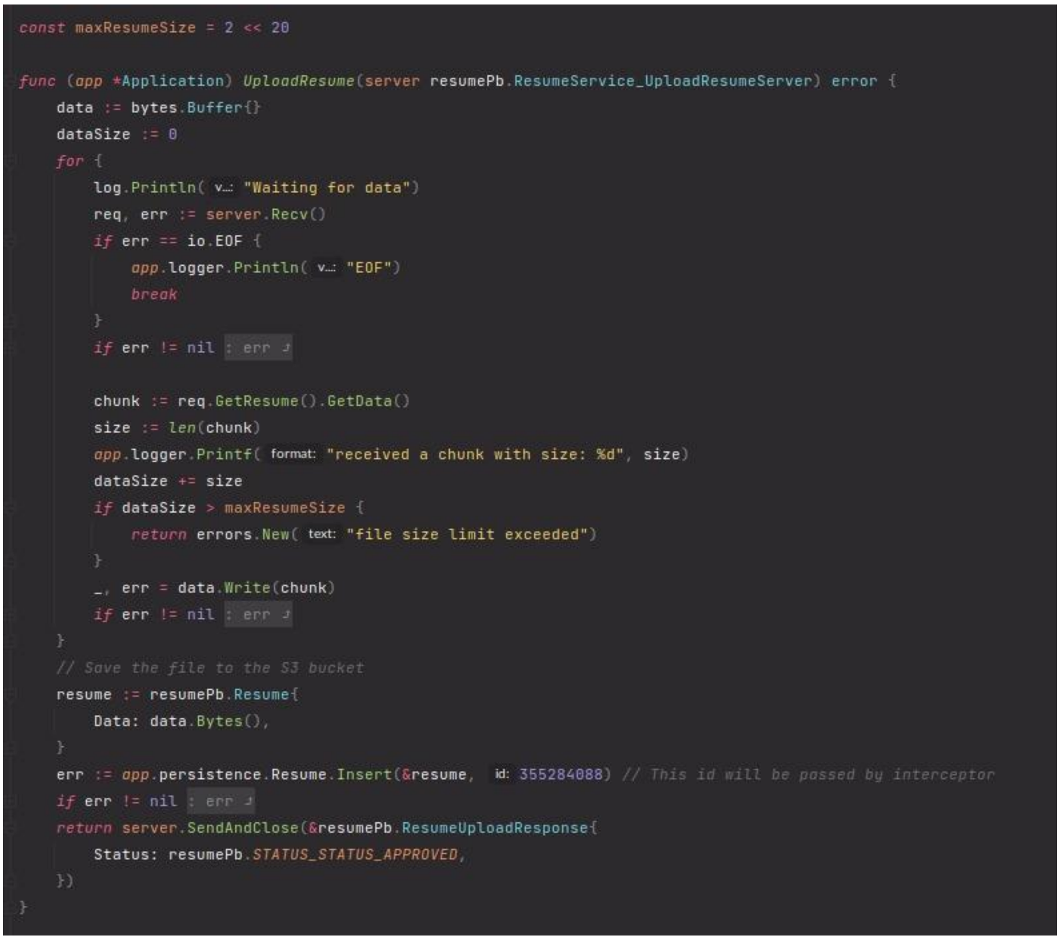
2. Unary Interceptor which intercepts the unary calls

`



```go
func (app *Application) Stream() grpc.StreamServerInterceptor {
    return func(
        srv interface{},
        stream grpc.ServerStream,
        info *grpc.StreamServerInfo,
        handler grpc.StreamHandler,
    ) error {
        log.Println( v...: "--> Stream Auth Interceptor", info.FullMethod)
        err := app.Authorize(stream.Context(), info.FullMethod)
        if err != nil {
            return err
        }
        return handler(srv, stream)
    }
}
```

3. This interceptor, intercepts the stream calls. Both unidirectional from server or client, and Bi-directional



```go
const maxResumeSize = 2 << 20

func (app *Application) UploadResume(server resumePb.ResumeService_UploadResumeServer) error {
    data := bytes.Buffer{}
    dataSize := 0
    for {
        log.Println( v...: "Waiting for data")
        req, err := server.Recv()
        if err == io.EOF {
            app.logger.Println( v...: "EOF")
            break
        }
        if err != nil : err ↲

        chunk := req.GetResume().GetData()
        size := len(chunk)
        app.logger.Printf( format: "received a chunk with size: %d", size)
        dataSize += size
        if dataSize > maxResumeSize {
            return errors.New( text: "file size limit exceeded")
        }
        _, err = data.Write(chunk)
        if err != nil : err ↲
    }
    // Save the file to the S3 bucket
    resume := resumePb.Resume{
        Data: data.Bytes(),
    }
    err := app.persistence.Resume.Insert(&resume,  id: 355284088) // This id will be passed by interceptor
    if err != nil : err ↲
    return server.SendAndClose(&resumePb.ResumeUploadResponse{
        Status: resumePb.STATUS_STATUS_APPROVED,
    })
}
```

4. Upload Resume function, which checks the bits and if file size exceeds then throws an error.

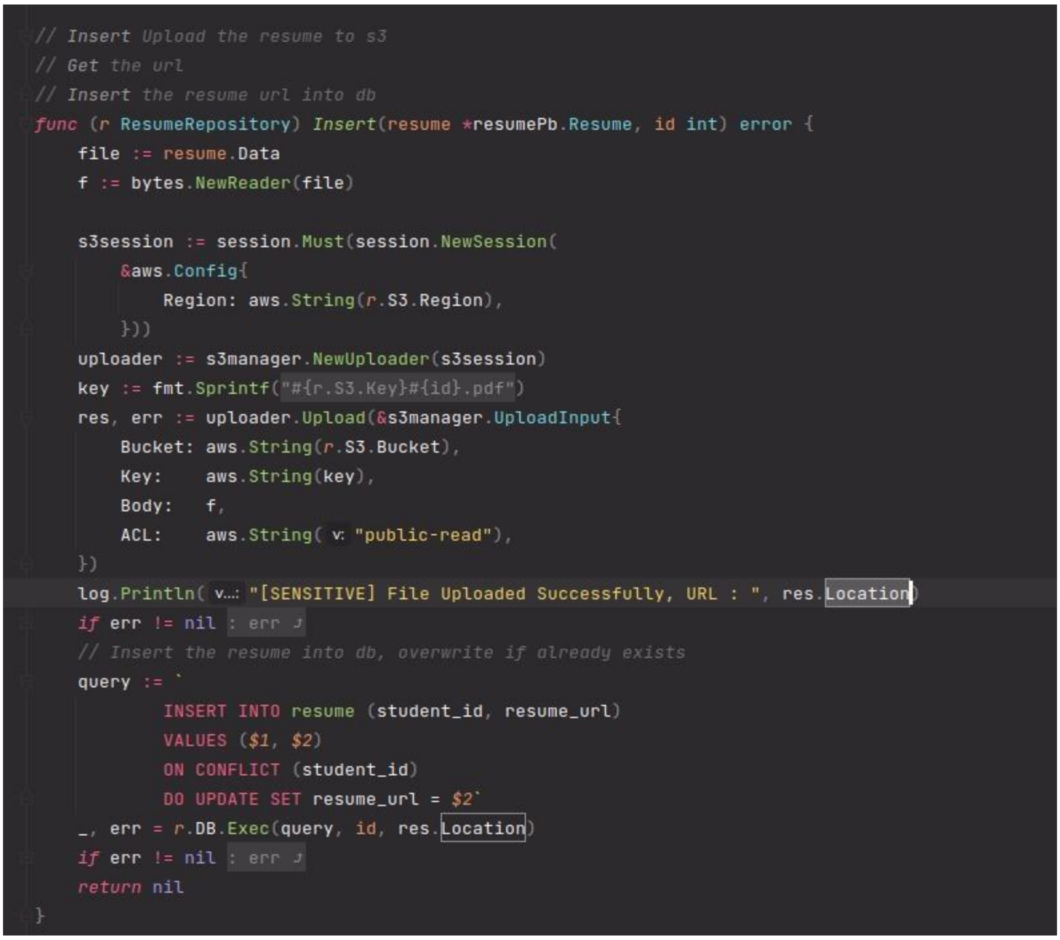
`

5. All database operations:

```
package persistence

import (
    jobApplicationPb "awesomeProject/internal/proto/application"
    authPb "awesomeProject/internal/proto/auth"
    hrPb "awesomeProject/internal/proto/hr"
    resumePb "awesomeProject/internal/proto/resume"
    studentPb "awesomeProject/internal/proto/student"
    "database/sql"
)

type Repositories struct {
    User interface {
        Insert(user *authPb.UserFields, hashedPassword string) error
        GetByUsername(username string) (*authPb.UserFields, error)
        DeleteByUsername(username string) error

        // Functions solely for auth
        Get(username string) (hashedPassword, role string, err error)
    }
    Student interface {
        // Please note that, id means here I'm refering to moodle id
        Insert(student *studentPb.Student) error
        Get(id string) (*studentPb.Student, error)
        Update(student *studentPb.Student) error
        Delete(id int64) error

        CheckProfileStatus(id string) (bool, error)
        GetGPA(id string) (*studentPb.Gpa, error)
        UpdateGPA(id string, gpa *studentPb.Gpa) error
    }
    Resume interface {
        Insert(resume *resumePb.Resume, id int) error
        Get(id int64) (string, error)
        Delete(id int64) error
    }

    Hr interface {
        Insert(hr *hrPb.Hr) (int, error)
```

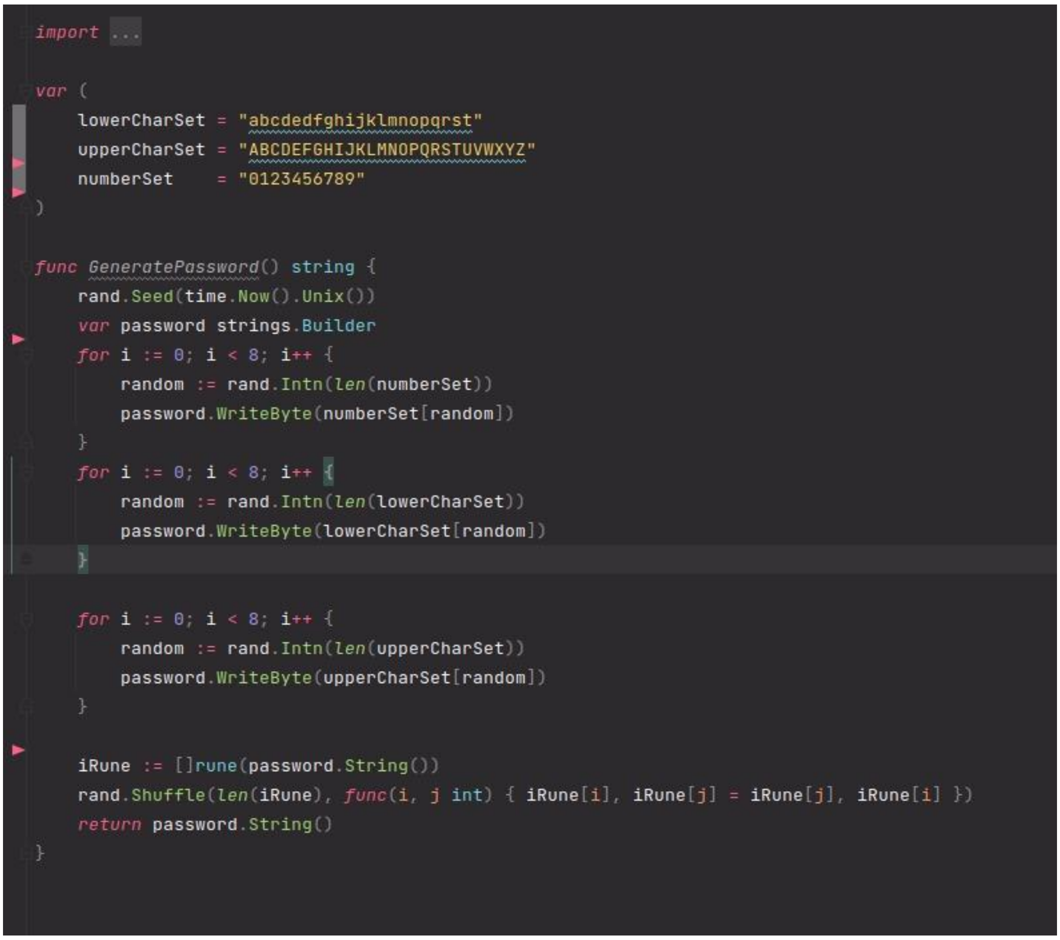
```
 `
        Get(id int64) ([]*hrPb.Hr, error)
        Update(hr *hrPb.Hr) error
        Delete(id int64) error
    }

    Job interface {
        Insert(hrId int, job *hrPb.JobFields) error
        Get(job *hrPb.JobSearchQuery) ([]*hrPb.JobFields, error)
        GetAll() ([]*hrPb.JobFields, error)
        Update(job *hrPb.JobFields) error
        Delete(id int64) error
        GetJobById(userId int) (*hrPb.JobFields, error)
    }
    JobApplication interface {
        Insert(userId, jobId int) error
        Get(applicationId int) (bool, error)
        GetAll(userId int)
([]*jobApplicationPb.AllJobApplicationStatus, error)
    }
}

func NewRepositories(db *sql.DB, s3 *S3) *Repositories {
    return &Repositories{
        User:           userRepository{DB: db},
        Student:         studentRepository{DB: db},
        Resume:           ResumeRepository{DB: db, S3: s3},
        Hr:             hrRepository{DB: db},
        Job:            jobRepository{DB: db},
        JobApplication: JobApplicationRepository{DB: db},
    }
}
```

`



```go
// Insert Upload the resume to s3
// Get the url
// Insert the resume url into db
func (r ResumeRepository) Insert(resume *resumePb.Resume, id int) error {
    file := resume.Data
    f := bytes.NewReader(file)

    s3session := session.Must(session.NewSession(
        &aws.Config{
            Region: aws.String(r.S3.Region),
        }))
    uploader := s3manager.NewUploader(s3session)
    key := fmt.Sprintf("#{r.S3.Key}#{id}.pdf")
    res, err := uploader.Upload(&s3manager.UploadInput{
        Bucket: aws.String(r.S3.Bucket),
        Key:    aws.String(key),
        Body:   f,
        ACL:    aws.String( v: "public-read"),
    })
    log.Println( v...: "[SENSITIVE] File Uploaded Successfully, URL : ", res.Location)
    if err != nil : err ↲
    // Insert the resume into db, overwrite if already exists
    query := `
        INSERT INTO resume (student_id, resume_url)
        VALUES ($1, $2)
        ON CONFLICT (student_id)
        DO UPDATE SET resume_url = $2`
    _, err = r.DB.Exec(query, id, res.Location)
    if err != nil : err ↲
    return nil
}
```

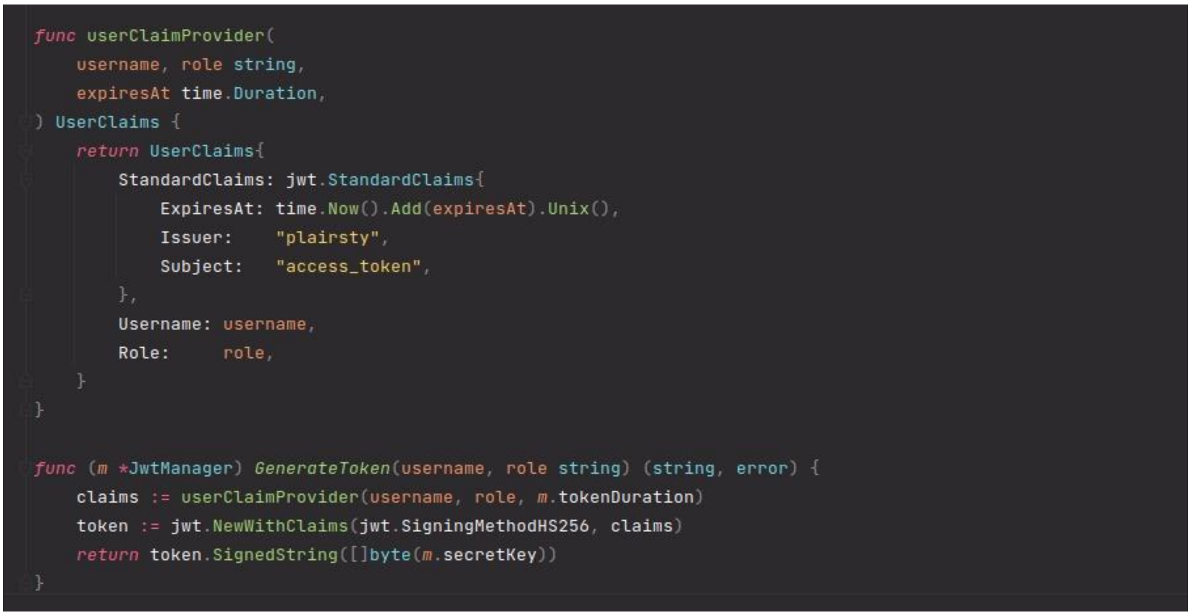
6. This function is required to insert the resume.



```go
import ...

var (
    lowerCharSet = "abcdedfghijklmnopqrst"
    upperCharSet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    numberSet    = "0123456789"
)

func GeneratePassword() string {
    rand.Seed(time.Now().Unix())
    var password strings.Builder
    for i := 0; i < 8; i++ {
        random := rand.Intn(len(numberSet))
        password.WriteByte(numberSet[random])
    }
    for i := 0; i < 8; i++ {
        random := rand.Intn(len(lowerCharSet))
        password.WriteByte(lowerCharSet[random])
    }

    for i := 0; i < 8; i++ {
        random := rand.Intn(len(upperCharSet))
        password.WriteByte(upperCharSet[random])
    }

    iRune := []rune(password.String())
    rand.Shuffle(len(iRune), func(i, j int) { iRune[i], iRune[j] = iRune[j], iRune[i] })
    return password.String()
}
```

7. This function generates the password.

`



```
func userClaimProvider(
    username, role string,
    expiresAt time.Duration,
) UserClaims {
    return UserClaims{
        StandardClaims: jwt.StandardClaims{
            ExpiresAt: time.Now().Add(expiresAt).Unix(),
            Issuer:    "plairsty",
            Subject:   "access_token",
        },
        Username: username,
        Role:     role,
    }
}

func (m *JwtManager) GenerateToken(username, role string) (string, error) {
    claims := userClaimProvider(username, role, m.tokenDuration)
    token := jwt.NewWithClaims(jwt.SigningMethodHS256, claims)
    return token.SignedString([]byte(m.secretKey))
}
```

8. This function helps in generating token.

`

Chapter No:9

RESULT AND DISCUSSION

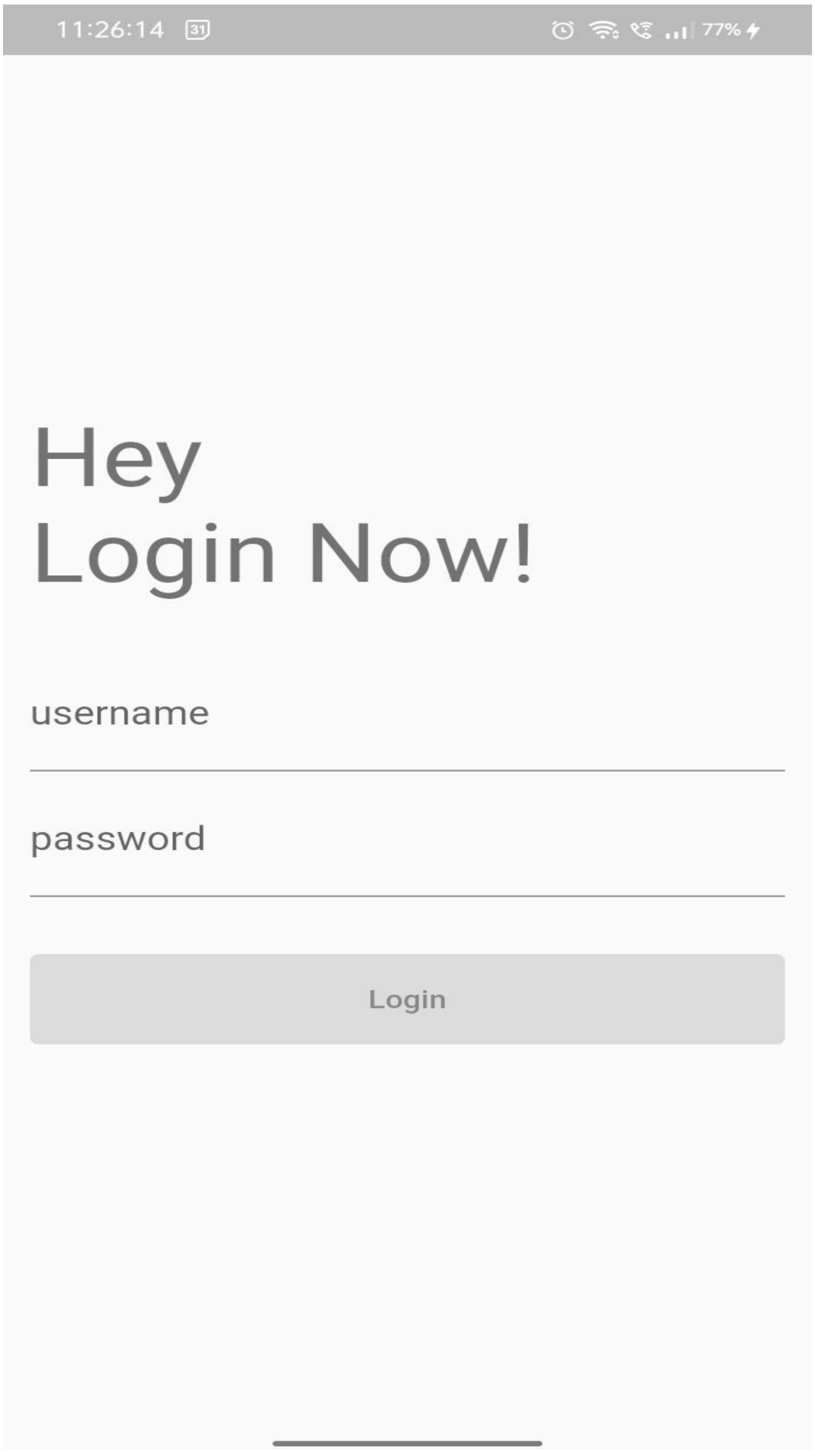
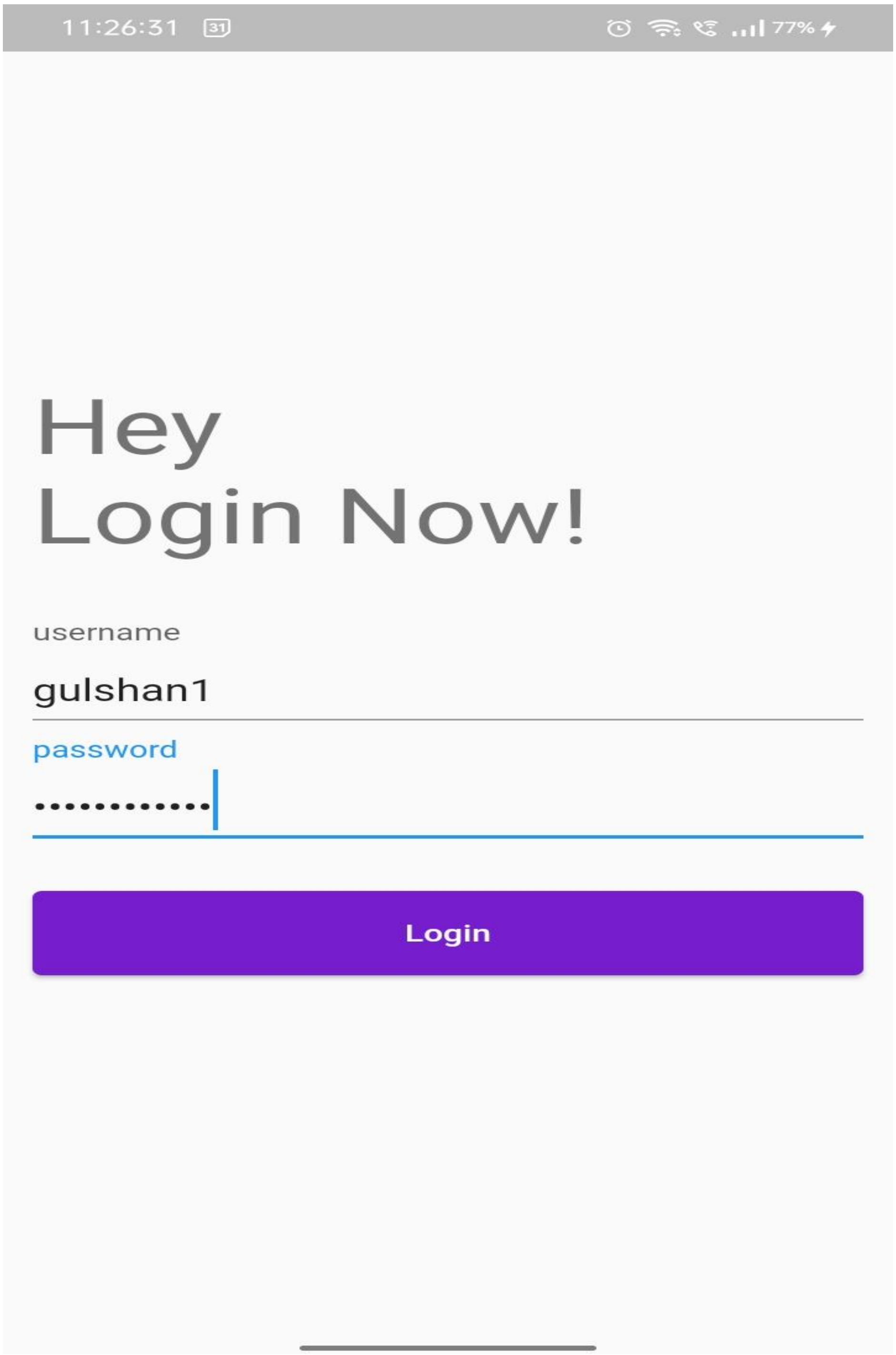


Fig 9.1: Login Page



Hey
Login Now!

username

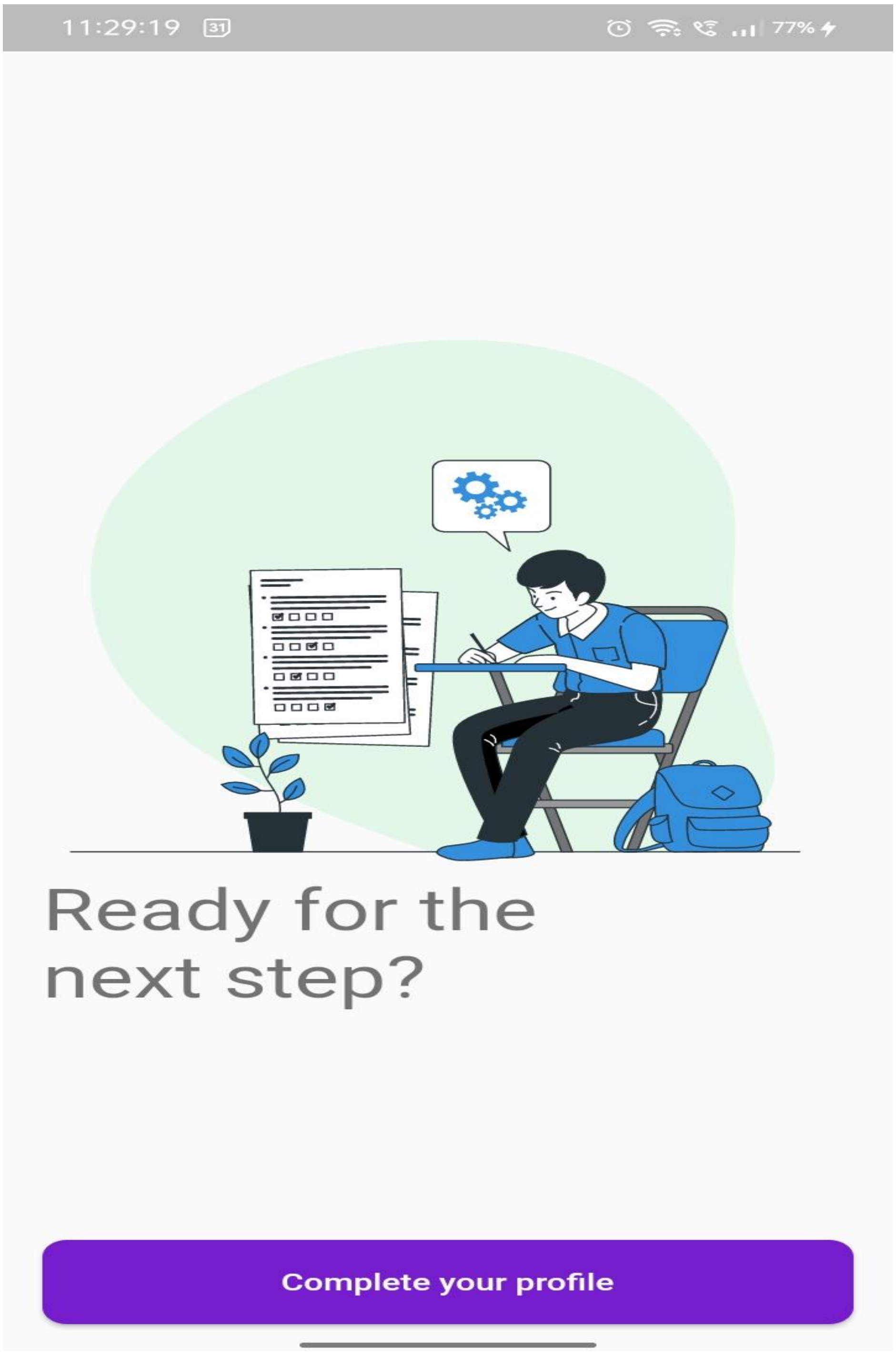
gulshan1

password

••••••••••••

Fig 9.2: User enters the details to login.

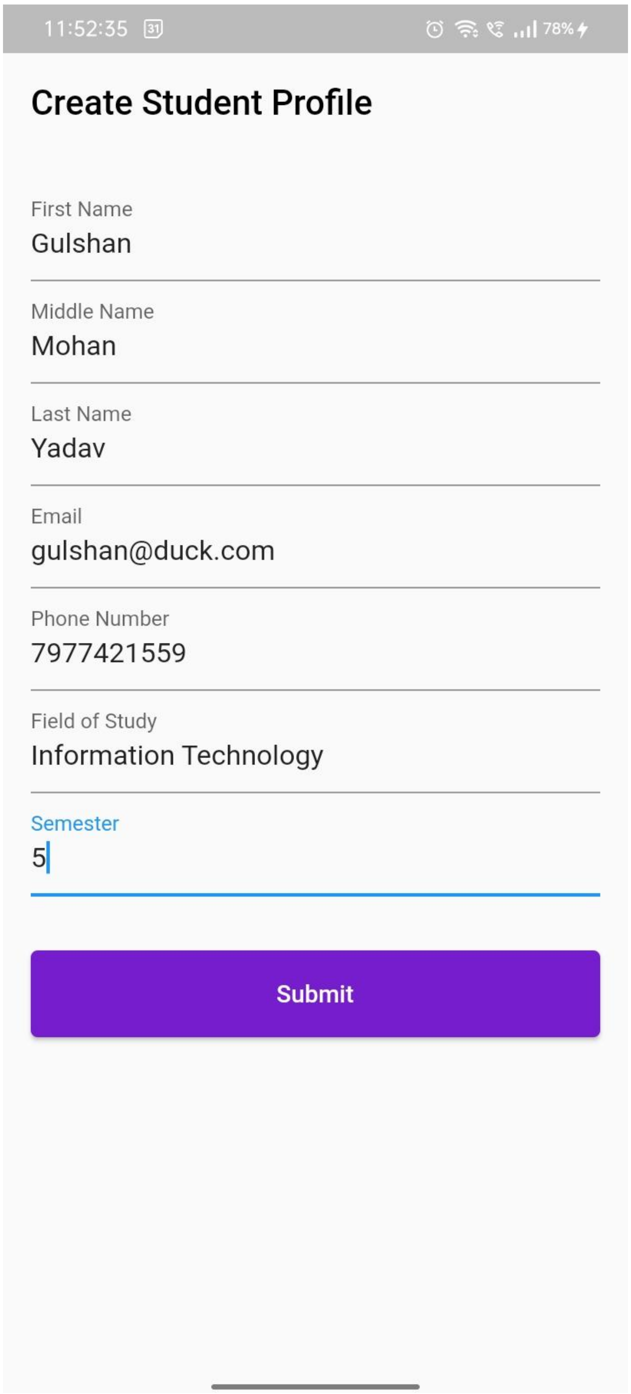
`



Ready for the next step?

Fig 9.3: The above page is displayed after logging in.

`



11:52:35 78%

Create Student Profile

First Name
Gulshan

Middle Name
Mohan

Last Name
Yadav

Email
gulshan@duck.com

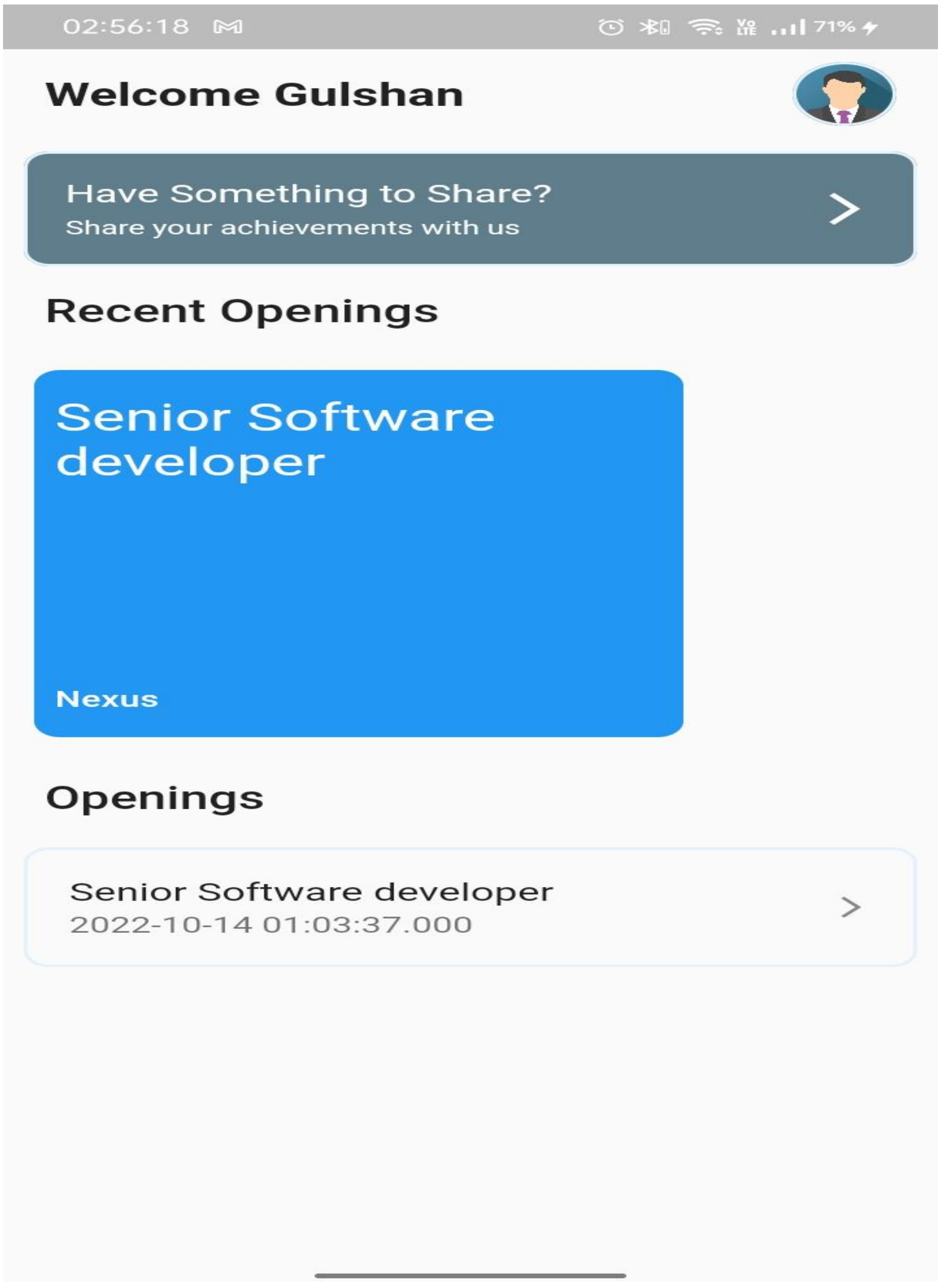
Phone Number
7977421559

Field of Study
Information Technology

Semester
5

Fig 9.4: Creating a Student Profile

`



02:56:18

71%

Welcome Gulshan

Have Something to Share?
Share your achievements with us

Recent Openings

Senior Software developer

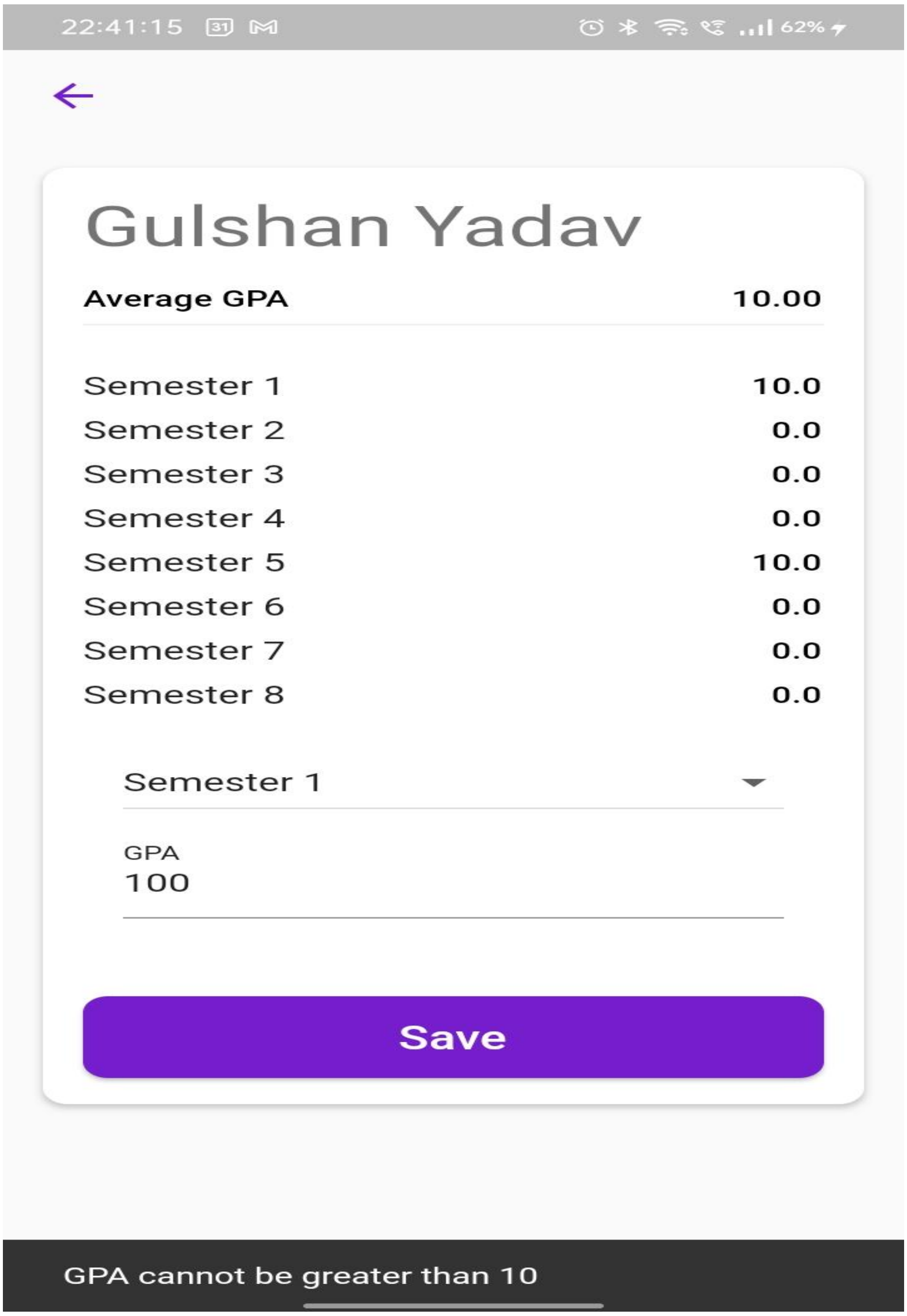
Nexus

Openings

Senior Software developer
2022-10-14 01:03:37.000

Fig 9.5: Dashboard

`



Gulshan Yadav

Average GPA	10.00
Semester 1	10.0
Semester 2	0.0
Semester 3	0.0
Semester 4	0.0
Semester 5	10.0
Semester 6	0.0
Semester 7	0.0
Semester 8	0.0

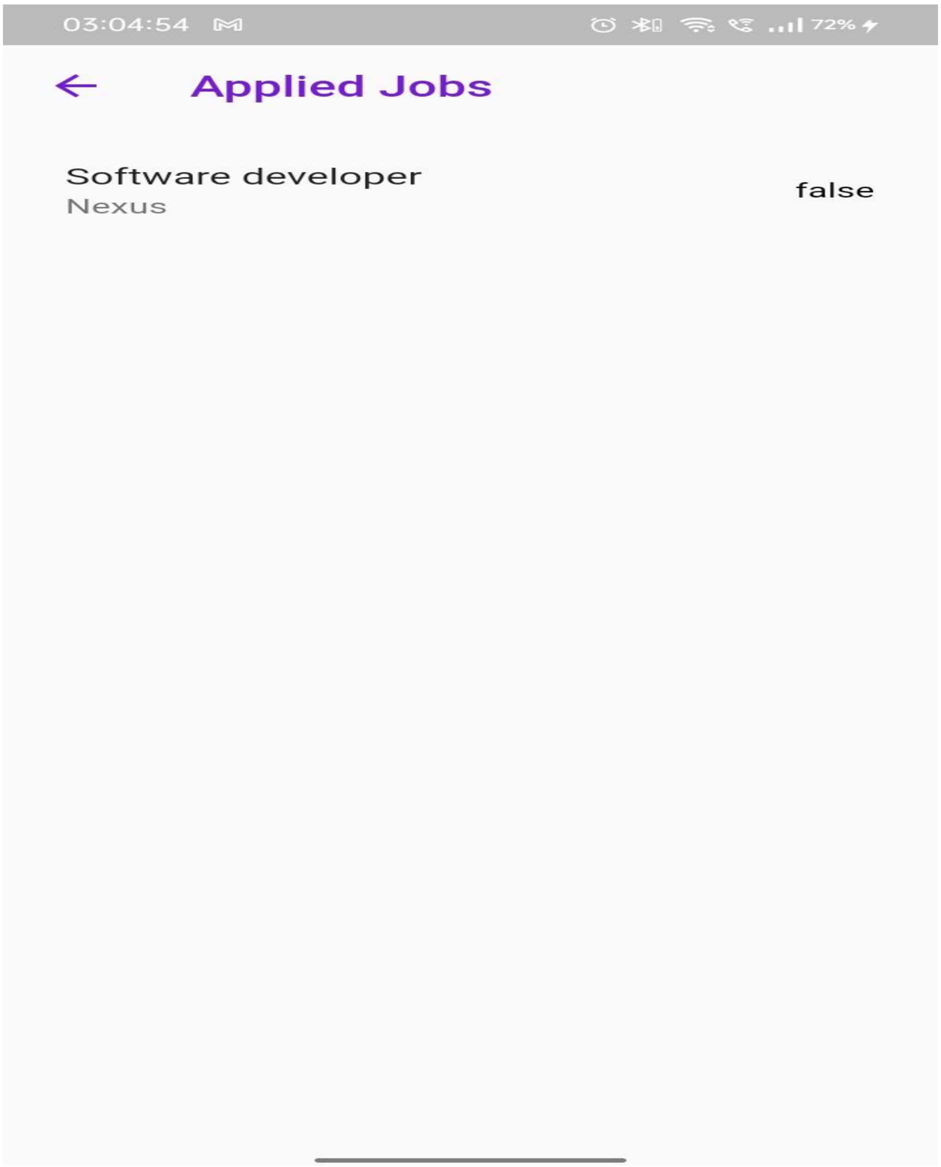
Semester 1

GPA
100

GPA cannot be greater than 10

Fig 9.6: Academics Page

`



Applied Jobs

Software developer
Nexus

false

Fig: 9.7: Uploading a Resume

`

Chapter No:10

CONCLUSION AND FUTURE SCOPE

Maximum work goes manually in the present placement system which makes it take time to avail changes. This includes main problems like searching for the data of students and sorting them along with it. Also,updating student data is a cumbersome job and does not have a method to notify the student in time which makes the management of the placements very difficult. In the proposed system, all of these problems become automated.The registration of the student for an upcoming placement,the addition of a new user, notifying students, sharing information, the privacy of the student, etc is all met. The admin validates the information and gives the student list based on the criteria required which otherwise would have been very difficult to manage.

Future scope of the project:
Though our project is itself matured enough but still betterment is always an open door. In this case also we can add some features to this software to make this software more reliable.
These are as follows:
Firstly, during the development of the project my prime object was to keep the hardware & software requirement as minimum as possible so that it supports maximum user base.
The searching procedure should be very strong like placement officer can search student as fast as possible.
Modify the project with better approach with more graphics.
The back-up procedure can be incorporated to make sure of the database integrity.
Recruiter can visit any time through this application and communicate with Placement officer.
Placement officer can contact with both student and company through message. Student and company can also send message to Placement officer.

`

Chapter No:11

References

1. https://www.tutorialspoint.com/flutter/index.html
2. https://www.w3schools.com/go/
3. https://www.postgresqltutorial.com/