MY PERSONAL E-DIARY AND SHEDULER

A Mini Project Report

submitted by

AHAMMED MUSHFIQ C P (MES22MCA-2003)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679582

December 2023

DECLARATION

I undersigned hereby declare that the project report **'MY PERSONAL E-DIARY AND SHEDULER'**, submitted for partial fulfillment of the requirements for the award of degree of Masterof Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of MR. Hyderali K, Head of Department, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:

Date:

AHAMMED MUSHFIQ C P(MES22MCA-2003)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **'MY PERSONAL E-DIARY AND SHEDULER'** is a bonafide record of the MiniProject work carried out by **AHAMMED MUSHFIQ C P (MES22MCA-2003)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

PG Coordinator HEAD OF THE DEPT

Acknowledgements

Our endeavor stands incomplete without dedicating our gratitude to a few people who have contributed towards the successful completion of our project. We pay our gratitude to the Almighty for His invisible help and blessing for the fulfillment of this work. At the outsetwe express our heart full thanks to our Head of the Department, Prof. HYDERALI K for permitting us to do this project. We take this opportunity to express our profound gratitude to Prof. HYDERALI K, our project guide for his valuable support. We also take this opportunityto thank our project coordinator MS. Febin Aziz for her timely advice and strict schedules to complete our project. We are also grateful to all our teaching and non-teaching staff fortheir encouragement, guidance and whole-hearted support. Last but not least, we are gratefully indebted to our family and friends, who gave us a precious help in doing our project

AHAMMED MUSHFIQ C P(MES22MCA-2003)

Abstract

“My Personal E-Diary and Scheduler ” is an advanced secure and easy use web-based application which let you write daily note, meetings, Schedule Events. The diary can be entered using typing or can also be written by using voice, by using voice to text feature,i.e the user doesn’t need to type the whole diary log they just need to speak what happened about the day and it will automatically writes the log and also the user doesn’t need to read the whole entry they can hear it by using the text to speech feature .
Use this application to organize your notes, Meetings, Events, and ideas. This application has a simple user interface which makes it easy for using, through this project it simplified and can easily manage your daily schedules and activities. And this web application is used by only the User. In this Python project, we will build a GUI-based “My Personal E-Diary” Project using Django and Database MySQL. This web application is easy to implement also it is user friendly so any one can simply perform it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

“My Personal E-Diary and Scheduler ” is an advanced secure and easy use web-based application which let you write daily note, meetings, Schedule Events, Remainders. The diary can be also written by using voice by using voice to text feature and also it can be read automatically using text to speech feature . Use this application to organize your notes, Meetings, Events, and ideas. This application has a simple user interface which makes it easy for using, through this project it simplified and can easily manage your daily schedules and activities. And this web application is used by only the User. In this Python project, we will build a GUI-based “My Personal E-Diary” Project using Django and Database MySQL. This web application is easy to implement also it is user friendly so any one can simply perform it.

1.1 Motivation

The "My Personal E-Diary and Scheduler" project is not just a technological endeavor; it represents a fusion of innovation and practicality aimed at revolutionizing the way individuals manage their daily lives. With its advanced features such as voice-to-text and text-to-speech capabilities, it offers a seamless and efficient experience for users who prefer both traditional typing and voice input methods. Developing a blog management system provides hands-on experience in various aspects of software development, including user interface design, backend development, database management. Also we need a Comment section that can use by the user without login so we can use that comment feature as a brainstorming method

1.2 Objective

The objective of this application is:

- **Develop a User-Friendly Interface:** Create a simple and intuitive user interface that allows users to easily navigate and interact with the application, ensuring a positive and seamless user experience.
- **Implement Voice-to-Text Functionality:** Integrate a voice-to-text feature that enables users to input diary entries effortlessly by speaking, promoting convenience and accommodating various user preferences.
- **Incorporate Text-to-Speech Capability:** Integrate a text-to-speech feature that allows users to listen to their diary entries, enhancing accessibility and providing an alternative means of interacting with the content..
- **Promote Educational Value:** Structure the project in a way that allows developers to gain insights into web development practices, database management, and the integration of voice recognition and synthesis technologies.
- **Optimize Application Performance:** Focus on optimizing the performance of the application to ensure fast loading times and a responsive user interface, contributing to an efficient and satisfying user experience.

1.3 Contribution

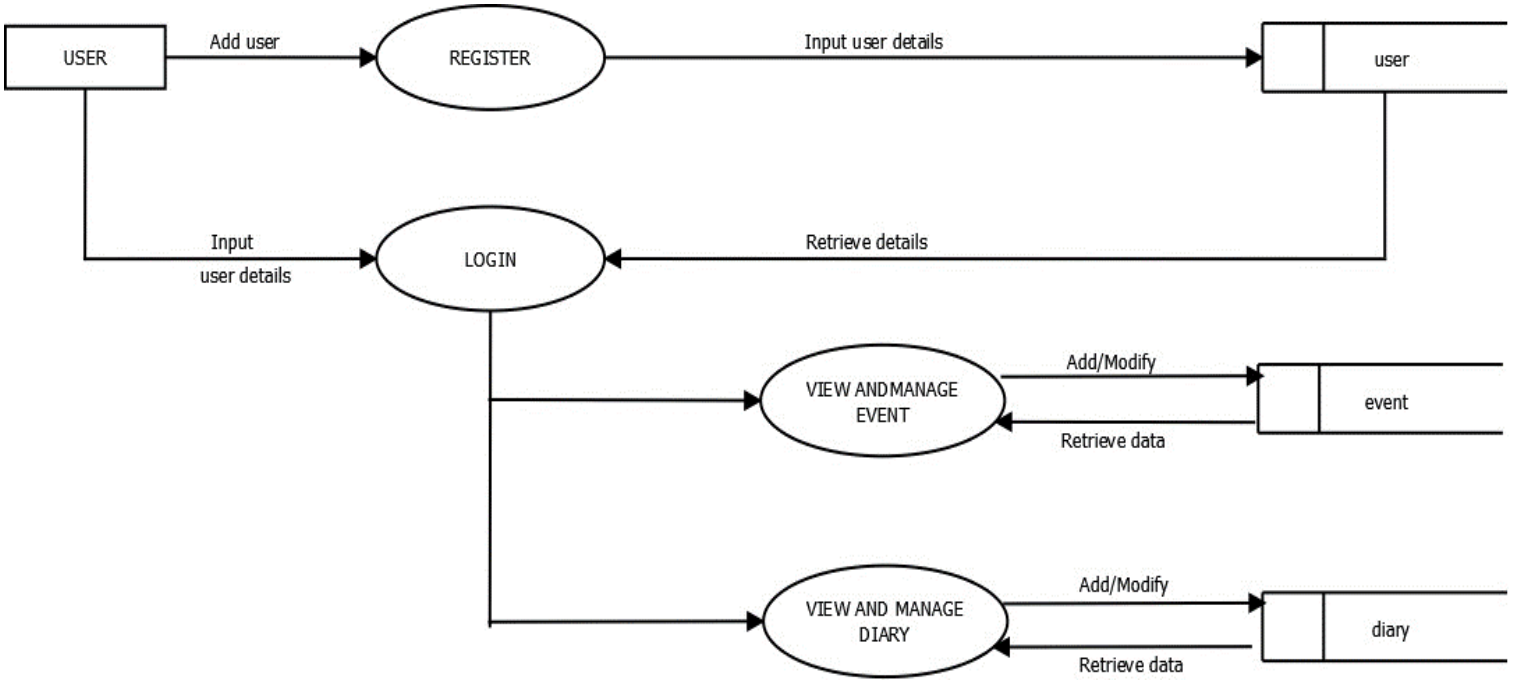
The major contribution in this project are : to develop a sophisticated and user-friendly web-based application that revolutionizes the traditional concept of diary-keeping and personal scheduling.

1.4 Report Organization

The project report is divided into three sections. Section 2: describes the methodology used for implementing the project. Section 3: gives the results anddiscussions. Finally, Section 4 gives the conclusion.

Data Flow Diagram - LEVEL 0

Data Flow Diagram - LEVEL 1(User)



Database Design

USER TABLE

Field Types

#	Field	Schema	Table	Type
1	id	ediary	auth_user	INT
2	password	ediary	auth_user	VARCHAR
3	last_login	ediary	auth_user	DATETIME
4	is_superuser	ediary	auth_user	TINYINT
5	username	ediary	auth_user	VARCHAR
6	first_name	ediary	auth_user	VARCHAR
7	last_name	ediary	auth_user	VARCHAR
8	email	ediary	auth_user	VARCHAR
9	is_staff	ediary	auth_user	TINYINT
10	is_active	ediary	auth_user	TINYINT
11	date_joined	ediary	auth_user	DATETIME

EVENT TABLE

Field Types

#	Field	Schema	Table	Type
1	id	ediary	cal_event	BIGINT
2	title	ediary	cal_event	VARCHAR
3	description	ediary	cal_event	TEXT
4	start_time	ediary	cal_event	DATETIME
5	end_time	ediary	cal_event	DATETIME
6	user_id	ediary	cal_event	INT

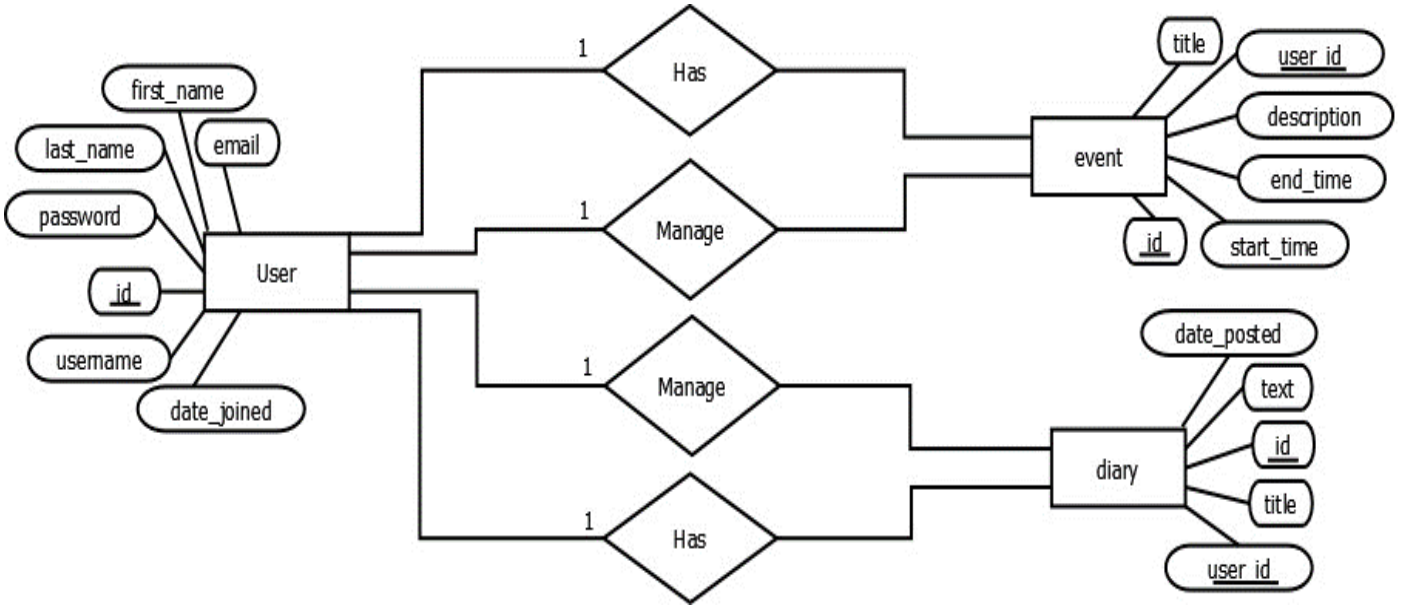
Database Design

DIARY TABLE

Field Types

#	Field	Schema	Table	Type
1	id	ediary	cal_entry	BIGINT
2	title	ediary	cal_entry	VARCHAR
3	text	ediary	cal_entry	TEXT
4	date_posted	ediary	cal_entry	DATETIME
5	user_id	ediary	cal_entry	INT

Entity Relationship Diagram



Chapter 2

Methodology

2.1 Introduction

After the initial studies it is found that agile model of software development is suitable and is the best method for the development of this system. Agile methodology mainly focused on the client satisfaction through continuous delivery. Also, it sets a minimum number of requirements and turns them in to a deliverable product. As this project has many individual requirements which can be delivered in parts and the user can gradually improve their work efficiency. Agile methodology has a family of methods of which scrum is selected for the development of this project. Scrum is process framework that has been used to manage complex product development. It is not a process or technique for building products rather it is a framework within which various processes can be employed. Also, it is suitable method to support the development process. It focuses on lean software development and has in building better software effectively and efficiently.

Agile is one of the most widely used and recognized software development frameworks. The methodology those experts agreed upon was described as 'lightweight' and fast. Agile is also about being the adaptive and continuous improvement, as much as it is about constant feedback and speed of delivery.

"Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iterations and evolves throughout the process by gathering feedback from the end users.

The major rules in scrum methodology are.

1. The product owner (PO): Who represents the stakeholder and the business.
2. The scrum master: Ensures the process followed, removes obstructions, and protects the development system.
3. Development team: Cross functional, self-organizing team who do the actual analysis, design implementation and testing process. They work together in iterative time boxed durations called sprints. The first step is the creation of the product backlog by the PO. It's a to-do list of stuff to be done by the scrum team. Then the scrum team selects the top priority items and tries to finish them within the time box called a sprint. An easier wayto remember all of this is to memorize the 3-3-5 framework. It means that a scrum project has3 roles,

3 artifacts,

and 5 events

These are: -

1. Roles: Product Owner, Scrum Master, and development team.
2. Artifacts: Product Backlog, Sprint Backlog and Product Increment.
3. Events: Sprint, Sprint planning, Daily Scrum, Sprint review and Sprint retrospective

The framework begins with a simple premise start with what can be seen or known. After that the progress is tracked and tweak as necessary. The three pillars of scrum are transparency, inspection, and adaptation. In scrum everyone has a role.

2.2 Module Description

The Major Modules are: -

- USER
  - LOGIN
  - REGISTER
  - DIARY
    - VIEW AND WRITE DAILY ACTIVITIES
      - WRITE BY TYPING
      - SPEECH TO TEXT
      - ADD PHOTO
    - READ DIARY
      - TEXT TO SPEECH
    - UPDATE DIARY
  - SCHEDULER
    - SCHEDULE DAILY ACTIVITIES
    - EDIT ACTIVITIES
    - REMINDER NOTIFICATIONS

2.3 User story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work. which drives collaboration, creativity, and a better product overall.

UserStory ID	As a <type of user>	I want to	So that I can
1	USER	Login	Login successful with correct Username and password
2	USER	WRITE DIARY	Record daily activities and add image
3	USER	Edit Diary	Edit the records
4	USER	Delete	Delete the records
5	USER	Schedule	Schedule Daily activities

2.4 Product backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. The product backlog is the single authoritative source for things that a teamworks on. That means that nothing gets done that isn't on the product backlog. Conversely, the presenceof a product backlog item on a product backlog does not guarantee that it will be delivered. Itrepresents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. TheScrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or canbe user-centric e.g. in the form of user stories

ID	PRIORITY	SIZE (Hours)	SPRINT	STATUS	NAME
1	Medium	8	1	Planned	Registration
2	Medium	4		Planned	Login
3	Medium	10	2	Partially Completed	Table Design
4	High	9		Planned	Coding
5	Medium	4	3	Planned	Testing data
6	High	6		Planned	Output generation

2.5 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it.

User Story ID	Sprint	Start Date	End Date	Day	Status
1	Sprint 1	15/09/2023	12/10/2023	20	Completed
2	Sprint 2	13/10/2023	03/11/2023	20	Completed
3	Sprint 3	08/11/2023	30/11/2023	21	Completed

2.6 Sprint backlog(plan)

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someoneon the team to complete.

This project has three sprints :-

SPRINT 1 (PLAN)

Backlog Item	Com-pletion Date	Original Estimat-e	Day 1 15/09	Day 2 20/09	Day3 21/09	Day 4 28/09	Day 5 29/09	Day 6 04/10	Day 7 05/10	Day 8 06/10	Day 9 11/10	Day10 12/10
		HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS
Form Design	15/09	2	2	0	0	0	0	0	0	0	0	0
Table Design	28/09	3	0	1	1	1	0	0	0	0	0	0
Coding	06/10	6	0	0	0	0	2	1	1	2	0	0
Testing & validation	12/10	2	0	0	0	0	0	0	0	0	1	1
Total		13	2	1	1	1	2	1	1	2	1	1

Table - 1

Backlog Item	Compl-etion Date	Original Estimate	Day 1 13/10	Day 2 18/10	Day 3 19/10	Day 4 20/10	Day 5 25/10	Day 6 26/10	Day 7 27/10	Day 8 01/11	Day 9 02/11	Day 10 03/11	Day 11 08/11
		HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS
Form Design	13/10	2	2	0	0	0	0	0	0	0	0	0	0
Table Design	18/10	1	0	1	0	0	0	0	0	0	0	0	0
Coding	01/11	9	0	0	1	2	1	1	2	1	1	0	0
Testing &validation	03/11	3	0	0	0	0	0	0	0	0	0	2	1
Total		15	2	1	1	2	1	1	2	1	1	2	1

Table -2

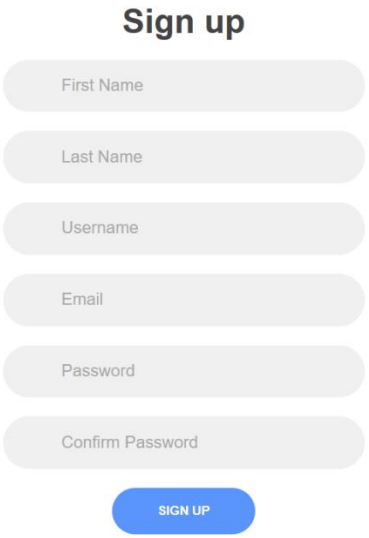
Backlog Item	Compl-etion Date	Original Estimate	Day 1 08/11	Day 2 09/11	Day 3 10/11	Day 4 15/11	Day 5 16/11	Day 6 17/11	Day 7 22/11	Day 8 23/11	Day 9 24/11	Day 10 29/11	Day 11 30/11
		HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS	HRS
Coding	22/11	10	1	2	1	1	2	1	1	1	0	0	0
Testing & validation	30/11	5	0	0	0	0	0	0	0	1	1	1	2
Total		15	1	2	1	1	2	1	1	2	1	1	2

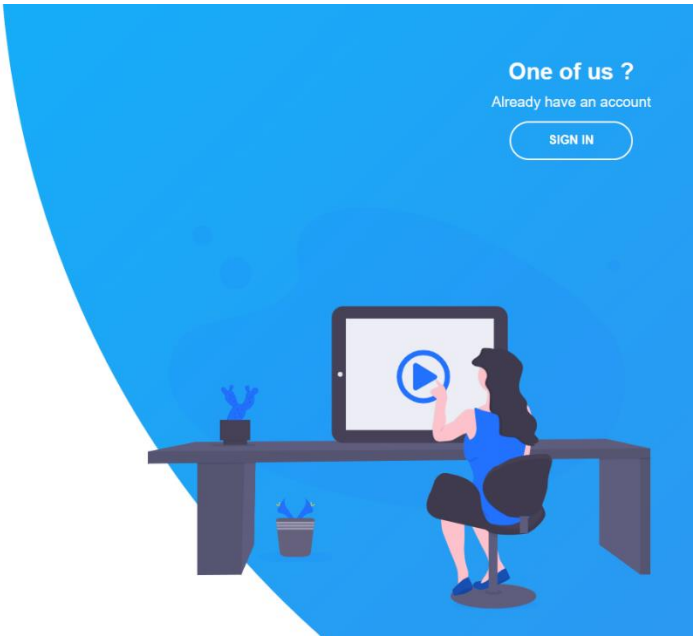
Table -3

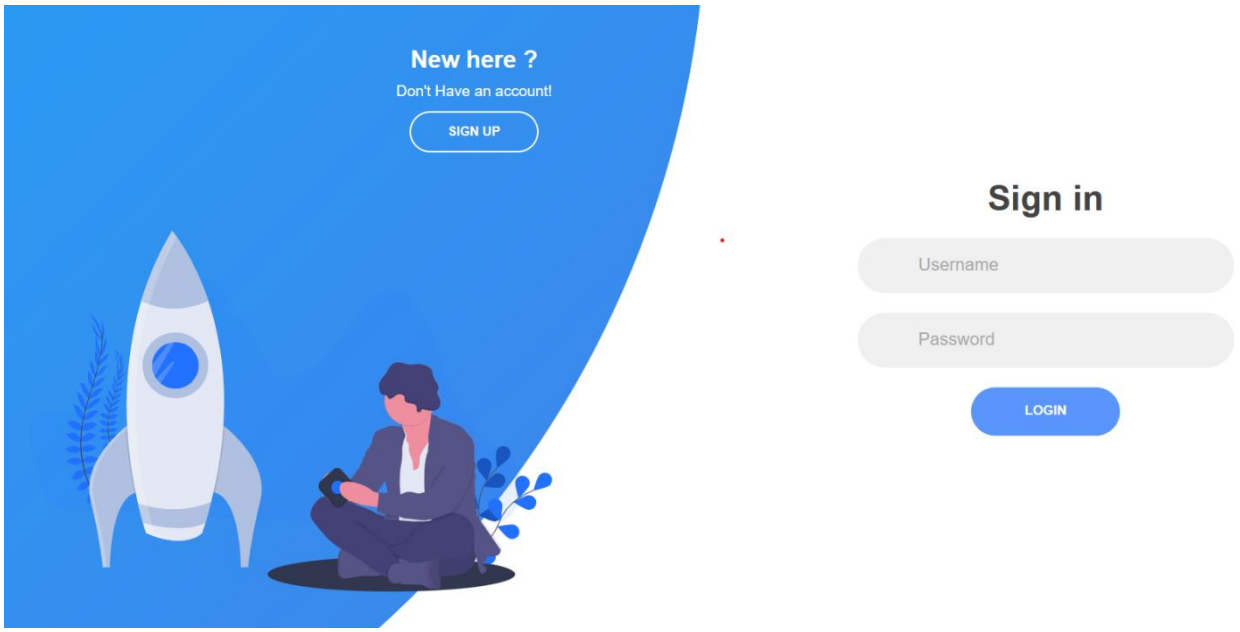
Chapter 3

Results and Discussions

3.1 Results







Event Form

Event Title:

Exam

Start Date:

12/04/2023 10:12 PM

End Date:

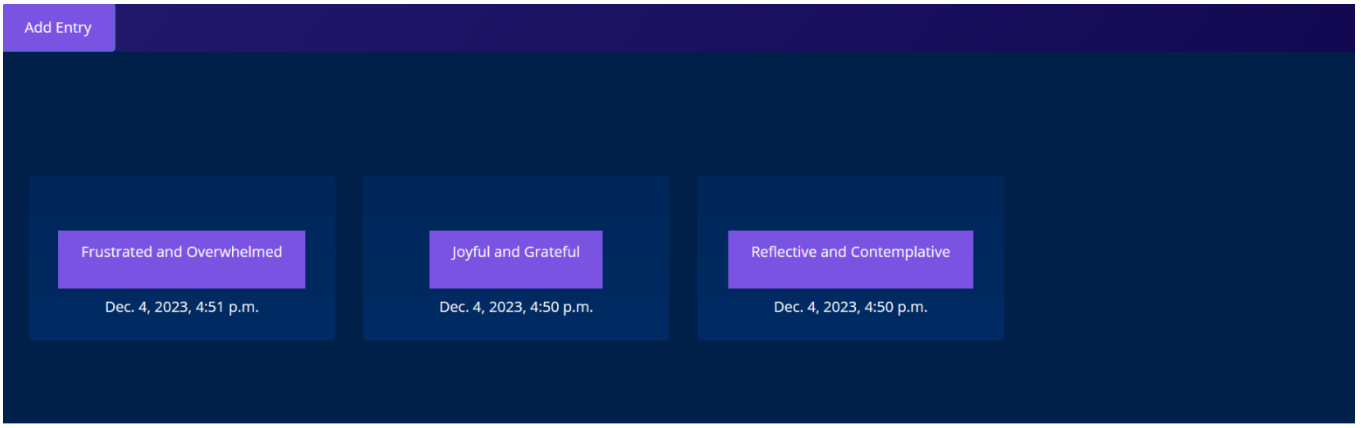
12/04/2023 10:13 PM

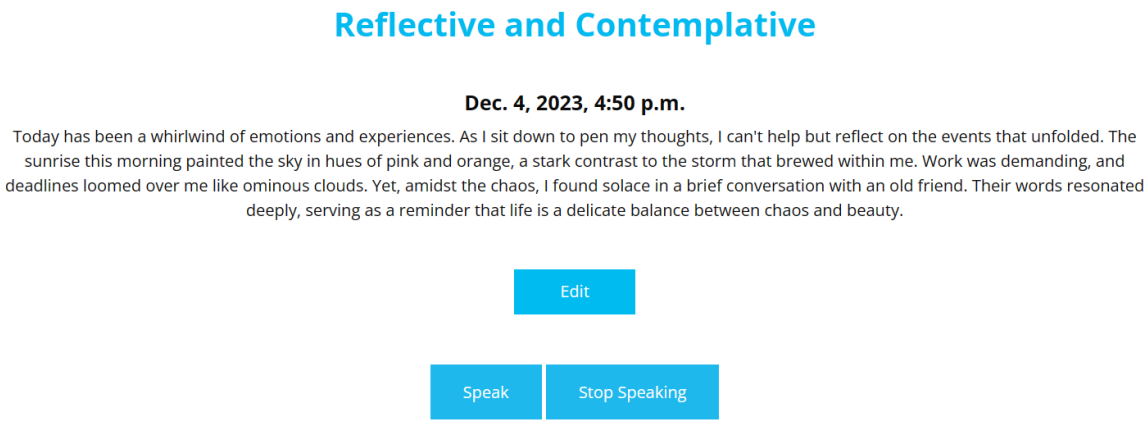
Event Description:

There is a Lab exam

Submit

December 2023						
Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
				1	2	3
4 • Exam	5	6	7	8	9	10 • Lab
11	12	13	14	15	16	17
18	19 • Meeting	20	21	22	23	24
25	26	27	28	29	30	31


Add Entry
Frustrated and Overwhelmed
Dec. 4, 2023, 4:51 p.m.
Joyful and Grateful
Dec. 4, 2023, 4:50 p.m.
Reflective and Contemplative
Dec. 4, 2023, 4:50 p.m.


Reflective and Contemplative
Dec. 4, 2023, 4:50 p.m.
Today has been a whirlwind of emotions and experiences. As I sit down to pen my thoughts, I can't help but reflect on the events that unfolded. The sunrise this morning painted the sky in hues of pink and orange, a stark contrast to the storm that brewed within me. Work was demanding, and deadlines loomed over me like ominous clouds. Yet, amidst the chaos, I found solace in a brief conversation with an old friend. Their words resonated deeply, serving as a reminder that life is a delicate balance between chaos and beauty.
Edit
Speak
Stop Speaking

Chapter 4

Conclusion

In conclusion, "My Personal E-Diary and Scheduler" is a sophisticated and user-friendly web-based application designed to streamline the organization of daily tasks, meetings, and notes. The integration of voice-to-text and text-to-speech features adds a layer of accessibility, making it convenient for users to input and retrieve information seamlessly.

The simplicity of the user interface enhances the overall user experience, ensuring that individuals, regardless of technical proficiency, can easily navigate and utilize the application. By leveraging the power of Django and MySQL, the project not only delivers a robust and scalable solution but also provides a secure environment for users to manage their personal data.

This project aims to simplify and enhance the daily lives of users by offering a centralized platform for note-taking, meeting scheduling, and event management. Whether it's jotting down thoughts in a diary, setting reminders for important tasks, or organizing meetings, "My Personal E-Diary and Scheduler" stands as a versatile tool for individuals seeking a more efficient and organized lifestyle.

By combining the power of technology with user-friendly design, this web application exemplifies how advancements in programming languages like Python, frameworks like Django, and databases like MySQL can come together to create practical solutions that cater to the needs of modern users. As technology continues to evolve, projects like these pave the way for a more connected and organized world, where managing daily schedules and activities becomes a seamless and enjoyable experience.

References

[1] https://studentprojectguide.com/php/alumni-management-system/.

[2] http://www.chegg.com/homework-help/questions-and-answers/project-name-alumni-managementsystem-project-project-details-hired-develop-software-track-q20206087

[3] https://www.studocu.com/in/document/goverment-womens-polytechnic-college-kalamassery/computer-science/alumni-management-system-report/31920708

Appendix

Source Code

```
from django.urls import path
from . import views

urlpatterns = [
    path('calendar', views.CalendarView.as_view(), name='calendar'),
    path('event', views.event, name='event_new'),
    path('event_edit/(?P<event_id>\d+)/$', views.editevent, name='event_edit'),
    path('diaryview', views.diaryview, name='diaryview'),
    path('adddiary', views.adddiary, name='adddiary'),
    path('view/<int:id>/', views.view, name='view'),
    path('view/<int:id>/editdiary/', views.editdiary, name='editdiary'),

]
```

```
import calendar
from pkgutil import get_data
from django.shortcuts import get_object_or_404, redirect, render
from datetime import date, datetime, timedelta
from django.http import HttpResponse, HttpResponseRedirect
from django.urls import reverse
from django.views import generic
from django.utils.safestring import mark_safe
from django.contrib.auth.decorators import login_required
from django.views.decorators.cache import cache_control
from django.utils.decorators import method_decorator

from .forms import EventForm

from .models import *
from .utils import Calendar
# Create your views here.
@method_decorator(cache_control(no_cache=True, must_revalidate=True, no_store=True),
name='dispatch')
class CalendarView(generic.ListView):
    model = Event
    template_name = 'calendar.html'

    def get_context_data(self, **kwargs):
        if self.request.user.is_authenticated:
            # print("asdf")
```

Appendix

```
# print(self.request.user)
            context = super().get_context_data(**kwargs)
            d = get_date(self.request.GET.get('month', None))
            print("====",self.request.user)
            cal = Calendar(d.year, d.month, self.request.user)
            html_cal = cal.formatmonth(withyear=True)
            context['calendar'] = mark_safe(html_cal)
            context['prev_month'] = prev_month(d)
            context['next_month'] = next_month(d)
            # print(context)
            return context
        else:
            return redirect('login')

def get_date(req_month):
    if req_month:
        year, month = (int(x) for x in req_month.split('-'))
        return date(year, month, day=1)
    return datetime.today()

def prev_month(d):
    first = d.replace(day=1)
    prev_month = first - timedelta(days=1)
    month = 'month=' + str(prev_month.year) + '-' + str(prev_month.month)
    return month

def next_month(d):
    days_in_month = calendar.monthrange(d.year, d.month)[1]
    last = d.replace(day=days_in_month)
    next_month = last + timedelta(days=1)
    month = 'month=' + str(next_month.year) + '-' + str(next_month.month)
    print(month)
    return month

# def event(request, event_id=None):
#     instance = Event()
#     if event_id:
#         instance = get_object_or_404(Event, pk=event_id)
#     else:
#         instance = Event()

#     form = EventForm(request.POST or None, instance=instance)
#     if request.POST and form.is_valid():
#         form.save()
#         return HttpResponseRedirect(reverse('calendar'))
#     return render(request, 'event.html', {'form': form})



@login_required
def event(request, event_id=None):
```

Appendix

```
    if request.method == 'POST':
        user=request.user
        title = request.POST['eventName']
        description = request.POST['eventDescription']
        start_time = request.POST['startDate']
        end_time = request.POST['endDate']
        event=Event.objects.create(user=user,title=title,description=description,start_
time=start_time,end_time=end_time)
        event.save
        return redirect('calendar')

    else:
        return render(request, 'eventss.html')
@login_required
def editevent(request, event_id):
    instance = Event.objects.get(id=event_id)
    if request.method == 'POST':
        instance.title = request.POST.get('eventName')
        instance.description = request.POST.get('eventDescription')
        instance.start_time = request.POST.get('startDate')
        instance.end_time = request.POST.get('endDate')
        instance.save()
        return redirect('calendar')
        # user=request.user
        # title = request.POST['eventName']
        # description = request.POST['eventDescription']
        # start_time = request.POST['startDate']
        # end_time = request.POST['endDate']
        # event=Event.objects.get(id=event_id)

    else:
        #event=Event.objects.get(id=event_id)
        return render(request, 'editevent.html', {"instance":instance})
@login_required
def adddiary(request):
    print(request.method)
    if request.method == 'POST':
        user=request.user
        title = request.POST['title']
        text = request.POST['entry']
        entry = Entry.objects.create(user=user,title = title, text=text)
        print(text)
        entry.save()
        return redirect('diaryview')
    else:
        return render(request, 'diary.html')




@login_required(login_url="login")
```

Appendix

```
def diaryview(request):
    entries = Entry.objects.filter(user=request.user).order_by('-date_posted')
    context = {'entries' : entries}
    return render(request, 'viewdiary.html' , context)

@login_required
def view(request,id):
    print(id)
    entry = Entry.objects.get(id=id)
    print(entry)
    context = {'entry':entry}
    return render(request, 'view.html' , context)
@login_required
def editdiary(request,id):
    instance = Entry.objects.get(id=id)
    if request.method == 'POST':
        instance.title = request.POST.get('title')
        instance.text = request.POST.get('entry')
        instance.save()
        return redirect(diaryview)

    return render(request, 'editdiary.html',{'instance':instance})
```