

# ATTENDANCE MANAGEMENT SYSTEM

(18CSC206J) SOFTWARE  
ENGINEERING AND PROJECT  
MANAGEMENT

UNIVERSITY PRACTICAL  
EXAMINATION



# TABLE OF CONTENTS

Problem Statement

Requirements

Design

Demo of 3 Modules

Test Cases

# PROBLEM STATEMENT

By maintaining the attendance manually, then efficient reports cannot be generated. The system can generate efficient weekly, consolidate reports based on attendance. As the attendances are maintained in registers it has been a tough task for admin and staff to maintain for a long time. Instead, the software can keep long and retrieve the information when needed.

This system developed will reduce the manual work and avoid redundant data. Attendance Management System basically has two main modules for proper functioning Admin module is has rights for creating any new entry of faculty and student details. User has the right to make daily attendance, generating reports. Attendance reports can be taken by given details of student details, date, and class.

The proposed system requires very little paperwork. All the data is fetched into the computer immediately and reports can be generated through computers. Moreover, work becomes very easy because there is no need to keep data on paper. Once entered it is locked by the teacher and the issue of cheating and forgery is eliminated. This system reduces time to verify the attendance and eliminates human error.

# REQUIREMENTS

## FUNCTIONAL REQUIREMENTS

- **Functional Requirements** can also be expressed in the form of “user story” which is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

## NON-FUNCTIONAL REQUIREMENTS

- **Non-functional requirement** (NFR) is a **requirement** that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

# FUNCTIONAL REQUIREMENTS

Requirement (#)	Requirement Specification	Department	Name of Business User	Status
E1FR1	Authentication of user whenever he/she logs into the Software user interface and should be safe	web dev Team & cyber security	Chinmaya Purohit	completed
E1FR2	The system should be flexible enough to edit the personal information according to the user.	web development Team	Thejaswin S	completed
E1FR3	The Software must be able to display the basic information of the user such as name, roll number, the number of leaves taken and required no. of days to be present for eligibility.	web dev and database management	AP Rohith	completed

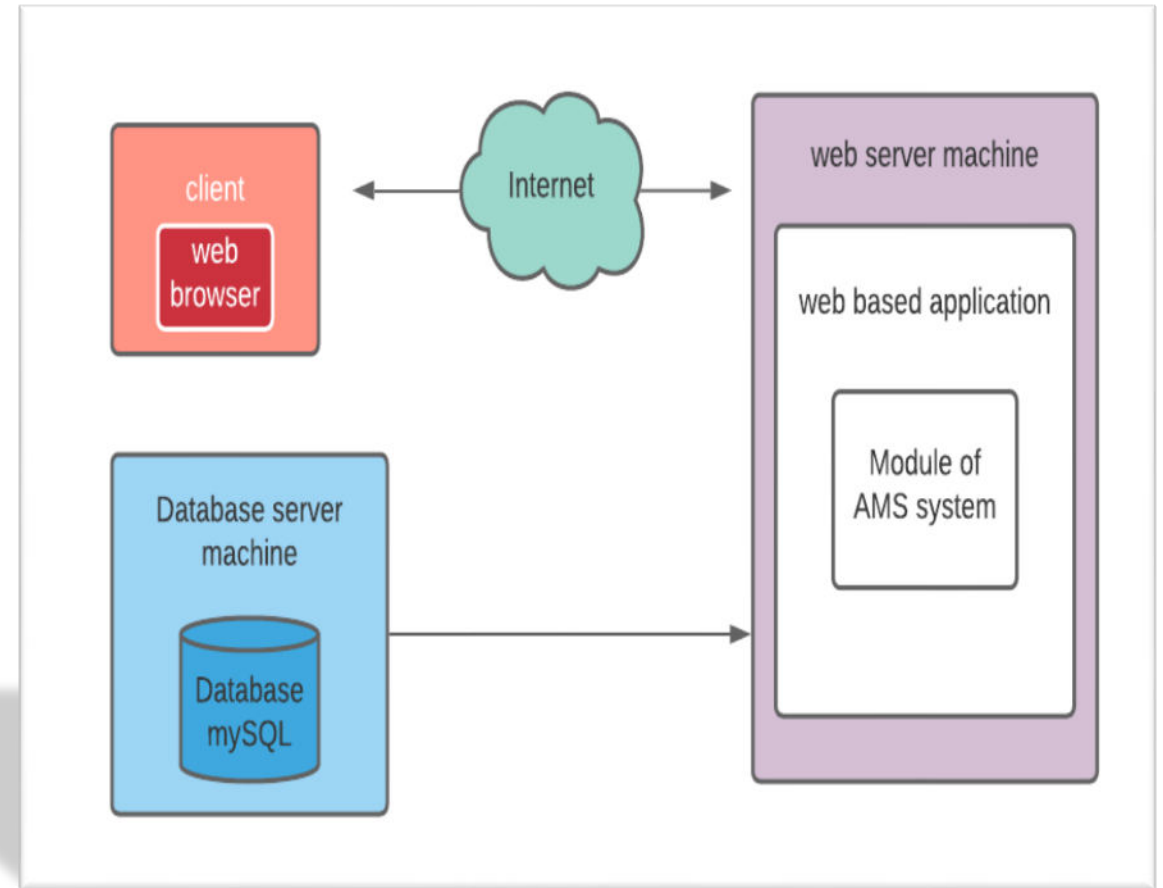
Functional Requirements		Requirement (#)	Category of NFR	Requirement Specification	Department	Name of Business User	Status
		NFR1	Performance	All pages should load with less time	Tech Department	Chinmaya Purohit	completed
		NFR2	Performance	Search should bring the results in less than 10 seconds	Tech Department	Thejaswin S	Completed
			Availability	Application should be available 24 hours and 7 days a week	Tech Department	AP Rohith	completed
			Scalability	Registration Service should scale to serve 1000 request per second over 5 minutes timespan	Tech Department	Chinmaya Purohit	completed
			Confidentiality	Only authorized users (students) belonging to a particular institution can have access to the software.	Tech Department	Thejaswin S	completed
		E1NFR2	Usability	When used for longer duration no issues are seen with respect to RAM consumption or any network problems	Tech Department	AP Rohith	completed
			Security	Data security of student's personal details and displaying what's required to the user.	Tech & cloud Department	Chinmaya Purohit	completed
			Traceability	User's data is untraceable	cyber security Department	Thejaswin S	completed
			Flexibility	Being able to view attendance whenever we try to login to the portal	Tech Department	AP Rohith	completed
Functional Requirements		E1NFR1	Extensibility	Ordering Service should be extensible to other countries within a week's time	Tech & marketing Department	Chinmaya Purohit	completed
			Interoperability	Ensuring the software works on different computers other than the host	Tech Department	Thejaswin S	completed
			Reliability	The quality of the software is trustworthy and its performance is consistent.	Tech & management Department	AP Rohith	completed
			Rapidity	To get the student's details in less time without any warnings or errors	Tech Department	Chinmaya Purohit	completed

# INFRASTRUCTURE REQUIREMENTS

Requirement (#)	Requirement Specification	Department	Name of Business User / Project Team Member	Status
IR1	Development Machine with min 4 GB Ram and 4 Cores	hardware development	system user	completed
IR2	Code Repository	web dev team	Thejaswin S	completed
IR3	UI of the AMS	web dev team	A P Rohith	completed
IR4	Proper databases and servers to manage and store data	web dev team	Chinmaya Purohit	completed
IR5	IDE-vscode	Technical team	AP Rohith	completed

# ARCHITECTURE DIAGRAM

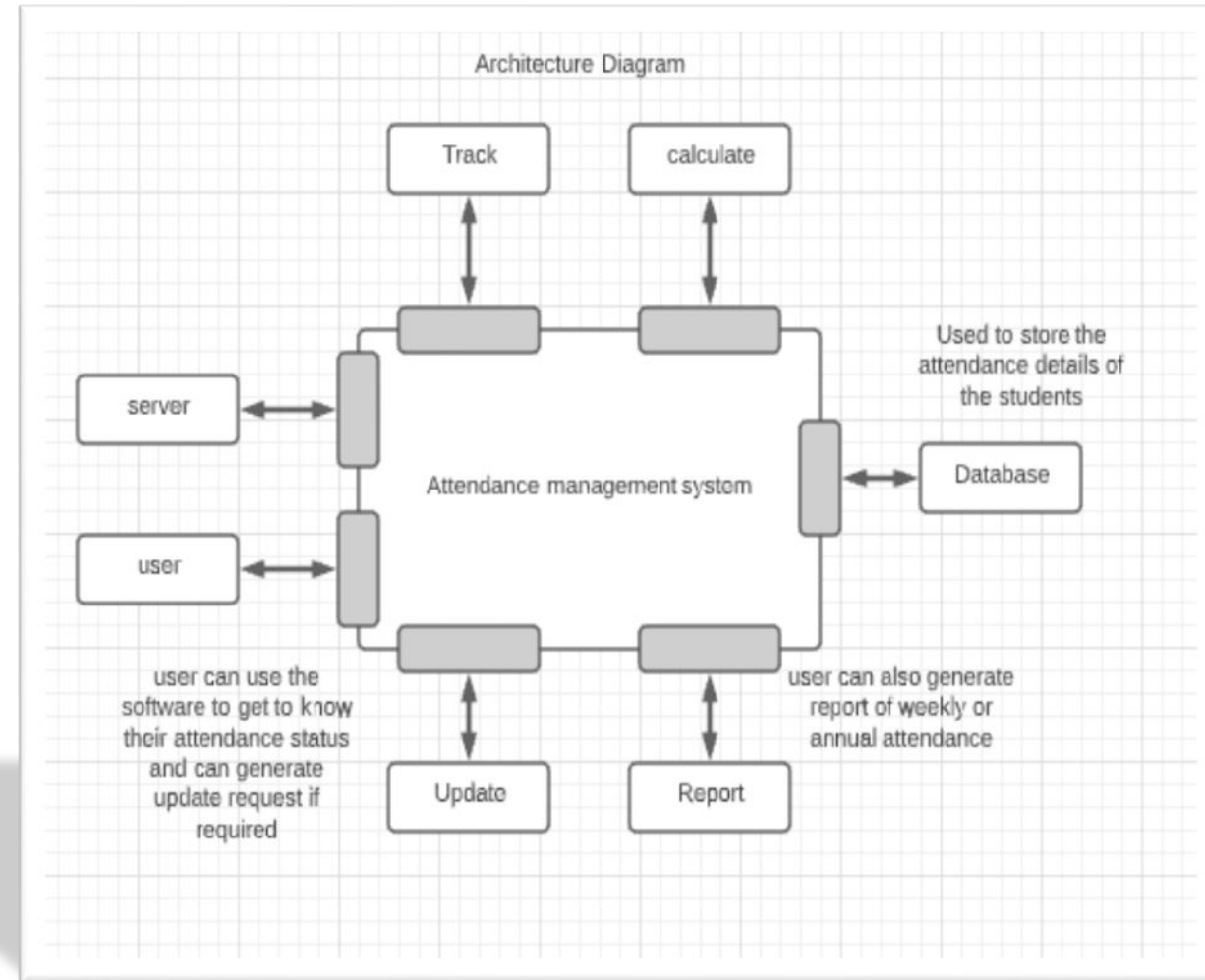
- An architecture diagram is a graphical representation of a set of concepts, that are part of the architecture, including their principles, elements, and components.



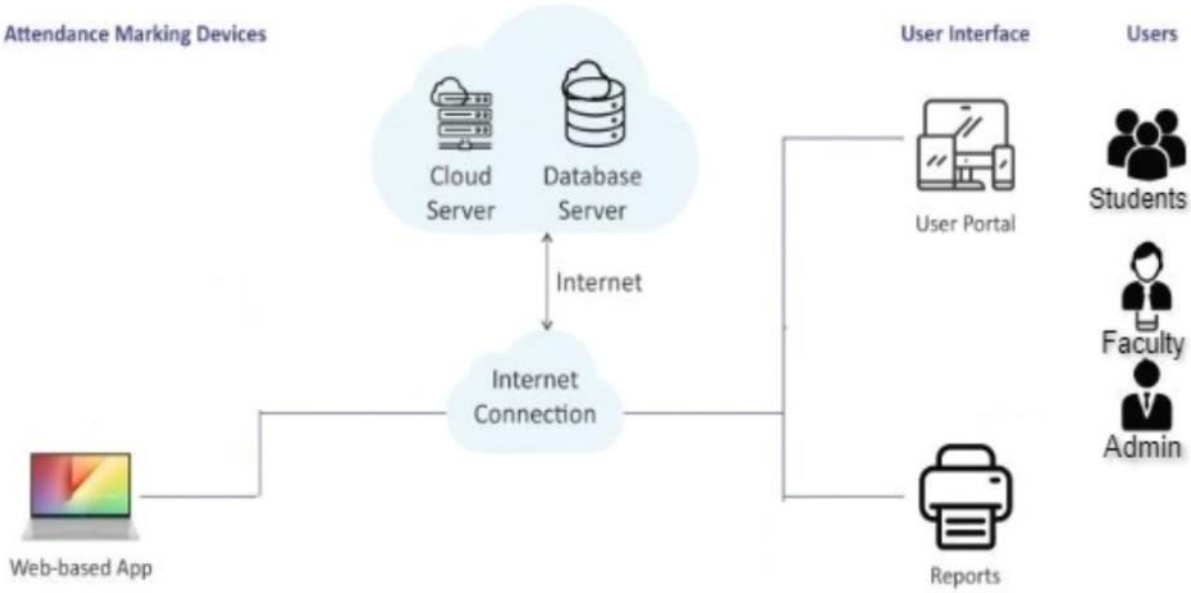


- The external entities that may involve in the project Attendance management system are:

- 1. User
- 2. Server
- 3.Track
- 4.Calculate attendance
- 5.Database
- 6.Update attendance
- 7.Report generation



### Attendance Marking Devices



### Presentation layer

Attendance Management System Portal

### Business Layer

Mark Attendance

View student list

View Report

### Persistence Layer

Database

# DEMO LOGIN PAGE

## Online Attendance Management System Login

Username

Password

Role

☒ Student ☐ Teacher ☐ Admin

Login

Have forgot your password? [Reset here.](#)

If you don't have any account, [Signup here](#)

# STUDENT PAGE

## Online Attendance Management System 1.0

[Home](#) [Students](#) [My Report](#) [My Account](#) [Logout](#)

Be attentive and be regular :)



## Online Attendance Management System 1.0

[Home](#) [Students](#) [My Report](#) [My Account](#) [Logout](#)

### Student List

Batch

[Go!](#)

Registration No.	Name	Department	Batch	Semester	Email
1	Pravesh	ECE	2019	4	rawatpravesh0016@gmail.com
2	Thejaswin S	CSE	2019	4	thejas@gmail.com
3	Chinmaya Purohit	CSE	2019	4	chinmay@gmail.com
4	AP Rohith	CSE	2019	4	rohith@gmail.com
5	ram	cse	2019	4	xyz@gmail.com

# Online Attendance Management System 1.0

[Home](#)[Students](#)[My Report](#)[My Account](#)[Logout](#)

## Update Account

Registration No.

Go!

Registration No.:

1

Student's Name:

Pravesh

Department:

ECE

Batch:

2019

Semester:

4

Email:

rawatpravesh0016@gmail.co

Update

Update

# Online Attendance Management System 1.0

[Students](#)[My Report](#)[My Account](#)[Logout](#)

## Student Report

Select Subject

Analysis of Algorithms

Your Reg. No.

enter your reg. no.

Go!

Registration No.:

1

Total Class (Days):

11

Present (Days):

8

Absent (Days):

3

# FACULTY'S PAGE

## Online Attendance Management System 1.0

[Home](#)[Students](#)[Faculties](#)[Attendance](#)[Report](#)[Logout](#)

### Student List

Batch (ex. 2019)

Registration No.	Name	Department	Batch	Semester	Email
1	Pravesh	ECE	2019	4	rawatpravesh0016@gmail.com
2	Thejaswin S	CSE	2019	4	thejas@gmail.com
3	Chinmaya Purohit	CSE	2019	4	chinmay@gmail.com
4	AP Rohith	CSE	2019	4	rohith@gmail.com
5	ram	cse	2019	4	xyz@gmail.com
6	ravi	ece	2019	4	ravi@qamil.com

# Online Attendance Management System 1.0

Home Students Faculties Attendance Report Logout

## Teacher List

Teacher ID	Name	Department	Email	Course
1	Sumit Bangar	cse	sumit@gmail.com	algo
12	kumar	cse	kumar@gamil.com	DAA



# Online Attendance Management System 1.0

[Home](#) [Students](#) [Faculties](#) [Attendance](#) [Report](#) [Logout](#)

Attendance of 2021-06-06

Enter Batch

Show!

Select Subject

Reg. No.	Name	Department	Batch	Semester	Email	Status
1	Pravesh	ECE	2019	4	rawatpravesh0016@gmail.com	Present <input checked="" type="radio"/> Absent <input type="radio"/>
2	Thejaswin S	CSE	2019	4	thejas@gmail.com	Present <input type="radio"/> Absent <input checked="" type="radio"/>
3	Chinmaya Purohit	CSE	2019	4	chinmay@gmail.com	Present <input checked="" type="radio"/> Absent <input type="radio"/>
4	AP Rohith	CSE	2019	4	rohith@gmail.com	Present <input type="radio"/> Absent <input checked="" type="radio"/>
5	ram	cse	2019	4	xyz@gmail.com	Present <input checked="" type="radio"/> Absent <input type="radio"/>
6	ravi	ece	2019	4	ravi@gamil.com	Present <input type="radio"/> Absent <input checked="" type="radio"/>

Save!

# Online Attendance Management System 1.0

[Students](#)[Faculties](#)[Attendance](#)[Report](#)[Logout](#)

## Individual Report

Select Subject

Student Reg. No.

## Mass Report

Select Subject

Date ( yyyy-mm-dd )

Student Reg. No:	1
Total Class (Days):	11
Present (Days):	8
Absent (Days):	3

# ADMIN'S PAGE

Online Attendance Management System 1.0

[Create Users](#) [Add Data](#) [Logout](#)

Create User

Email

your email

Username

choose username

Password

choose a strong password

Full Name

your full name

Phone Number

your phone number

Role

☒ Student ☐ Teacher ☐ Admin

Signup

Already have an account? [Login](#) here.

# Online Attendance Management System 1.0

[Create Users](#)[Add Data](#)[Logout](#)

Select: Teacher | Student

## Add Student's Information

Reg. No.	<input type="text" value="student reg. no."/>
Name	<input type="text" value="student full name"/>
Department	<input type="text" value="department ex. CSE"/>
Batch	<input type="text" value="batch ex 2020"/>
Semester	<input type="text" value="semester ex. Fall-15"/>
Email	<input type="text" value="valid email"/>

[Add Student](#)

## Add Teacher's Information

Teacher ID	<input type="text" value="teacher's id"/>
Name	<input type="text" value="teacher full name"/>
Department	<input type="text" value="department ex. CSE"/>
Email	<input type="text" value="valid email"/>
Subject Name	<input type="text" value="subject ex. Software Engineering"/>

[Add Teacher](#)

# TEST CASE

## Functional Testing

**Functional testing** verifies each **function**/feature of the software

Functional testing can be done manually

Functional testing is based on customer's requirements

A Functional Testing example is to check the login functionality

## Non-Functional Testing

**Non Functional testing** verifies non-functional aspects like performance, usability, reliability, etc.

Non Functional testing is hard to perform manually.

Non Functional testing is based on customer's expectations.

a Non Functional testing example is to check the dashboard should load in 2 seconds.

# FUNCTIONAL TEST CASES

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1.	Verify User Login	Accept Valid Username and Password on the Page#1 and not accept empty fields.	<ol style="list-style-type: none"> <li>User choose the category based on student, teacher, admin</li> <li>User clicks on text area of username and password</li> <li>Enter the username and password on the text box</li> <li>Click Login button</li> </ol>	User should be taken to the next page to use various features and check details and if left fields blanks and message should be shown to fill the details.	User is directed to another page which contains several options and if fields left blank, message is shown.	Pass	success
2.	Add User Registration and sync of user details on the database.	Accept email, username and phone number on the registration page	<ol style="list-style-type: none"> <li>User clicks on signup</li> <li>User enters the required details.</li> <li>User clicks Signup button</li> </ol>	User should be directed to registration page and after adding successfully login option is shown and data should be added to database.	User directed to registration page on clicking signup and can go back to login page on clicking login and data is updated in database.	Pass	Success
3.	Verify user registration	Not Accept empty fields for registration	<ol style="list-style-type: none"> <li>User clicks on signup</li> <li>User not enters the required details.</li> <li>User clicks Signup button</li> </ol>	Message should be shown to tell the user enter details instead of leaving blank.	Message is shown that fields cannot be empty	Pass	Success

# FUNCTIONAL TEST CASES

4.	Verify user registration	Accept valid email and phone number	<ol style="list-style-type: none"> <li>1. User clicks on signup</li> <li>2. User enters the invalid details.</li> <li>3. User clicks Signup button</li> </ol>	Prompt message to tell user invalid.	Accept invalid mail and phone number.	Fail	NIL
5.	Updating details	Change of details if provided incorrect previously.	<ol style="list-style-type: none"> <li>1. Click my account and reg. Number.</li> <li>2. Enter correct details</li> <li>3. Click update</li> </ol>	Should be able to re-enter the details again and updated details should get reflected in database.	Students can update and make changes in details by entering again and update is done in database.	Pass	Success
6.	Update attendance on daily basis of each subject	Select present and absent option for each student based on the presence of students in class daily.	<ol style="list-style-type: none"> <li>1. Select attendance option in teacher's page and enter the date and subject in respective fields.</li> <li>2. Click present or absent.</li> <li>3. Click save.</li> </ol>	Faculties will be able to smoothly enter attendance daily and is updated in database.	Faculties can enter present or absent and update the attendance on daily basis for each subject.	Pass	Success

# FUNCTIONAL TEST CASES

7.	Attendance Report	View and display individual report	<ol style="list-style-type: none"> <li>1. Click report option in student's or teacher's page</li> <li>2. Select subject and enter reg no.</li> <li>3. Click Go</li> </ol>	Able to display and view the attendance stats eg. No. Of days absent, present, class.	Can display and view the attendance of student by both students and teachers on the website	Pass	Success
		View and display Mass report	<ol style="list-style-type: none"> <li>1. Click report option in teacher's page</li> <li>2. Select subject and enter reg no.</li> <li>3. Click Go</li> </ol>	Should be able to view mass report of students	Not able to view and display mass report.	Fail	NIL
8.	Logout	Logout out of the session and account.	<ol style="list-style-type: none"> <li>1. click log out option.</li> </ol>	User should be able to logout from their accounts after using the website.	User can logout from the account in website by clicking logout.	Pass	Success



# NON-FUNCTIONAL TEST CASES

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1.	Performance	Computed average time to operate program (150 Trails)	1. Open the website and run it. 2.Do the same for 100 Trails and calculate the average time taken	Less Than 6 sec of computing Time	Approximately less than or equal to 3 sec	Pass	Success
2.	Availability	Program Should be able to run for more than 24 hours continuously without Crashing.	Open website and run the program for around 45 hours	Program should work without Crashing for more than 24hrs	Program worked with no problem for up to 45 hours	Pass	Success
3.	Scalability	Program should not crash when more than 1000 logins or 1000 request per second over 5mins time span	1.Open website in multiple devices at the same time 2.Try logging in at the same time	Program should not crash	Program crashed at approximately 500 logins and requests	Fail	Failure

# NON-FUNCTIONAL TEST CASES

4.	Usability	When used for longer duration no issues are seen with respect to RAM consumption or any network problems	1. Open the website and run it. 2.Measure the CPU performance and temperature.	CPU usage should be minimized	Optimum CPU usage recorded	Pass	Success
5.	Security	Website should be secured form wrong Logins, Glitches and Data Leaks	1.Alpha version of Program is given to Testers 2.Testing for glitches and Data Leaks performed	Program Should pass at least 98% Of total trails	Program Executed without Glitches and without security Breaches for 99.9%	Pass	Success
6.	Flexibility	User should be able to get the expected result and not directed to many pages when clicked different options and able to change easily in response to different user	1. Allow set of people to use the website and get their feedback. 2. check manually all the options available in software.	Website should be comfortable to use and easily accessible.	Flexibility in software is assured.	Pass	Success





*Thank  
you*