Slot : L45+L46

Team number : 06

Register number : 19BCE2074

Name : Kulvir Singh

Phase 1 – Requirement Analysis (Modified)

According to the current need to manage student, teacher and parent data, the following requirements were encountered:

Admin is required to create the presence of the coaching institute on the database. Student data which would include the personal details of the student. The teacher data and the parent data would be required. Fee structure incorporated by the admin for the course is required. Subjects such as physics, biology, mathematics and chemistry and the reference books along with topics are required. The marks for the aforementioned subjects were required. Lastly a tab on student attendance is required.

The relationship of the collected data is :-

The admin would be responsible for administrating the all the entities and will have all access and perform supervisory functions on all entities. The parents would act as guardian of the student and will have access to certain data related to the student. The student will be related to the subjects according to the type of his coaching, i.e., JEE or NEET. The student attending a class will be related in the attendance table. The fees being paid or not will be related to the parent and student having paid the fees or not. The teacher, student and parent will be able to marks. The teacher teaches all the above mentioned subjects.

With the pandemic bringing the normal to a standstill, people have shifted their business online. This enforced digitization has hampered and favored some organizations. Coaching Institutions across India which cater to more than a million of aspirants have found it difficult to carry on the same influx of students with this pandemic situation. Some big players in this industry have their own database management system, however some small and upcoming coaching institutes are finding it difficult to manage the student data.

This open source project helps the small scale institutes by providing them with a free template of a database management system required to run a coaching center in India.

***Entity sets and its attributes*:**

Strong Entities:

Coaching(**coaching_code**,coaching_name,city,address,contact)

Admin(**username**,password)

Biology(**Sno**,chapter,book)

Mathematics(**Sno**,chapter,book)

Physics(**Sno**,chapter,book)

Chemistry(**Sno**,chapter,book)

Student(**username**,password,gender,address,first_name,last_name,contact)

Parent(**username**,password,first_name,last_name,address,contact)

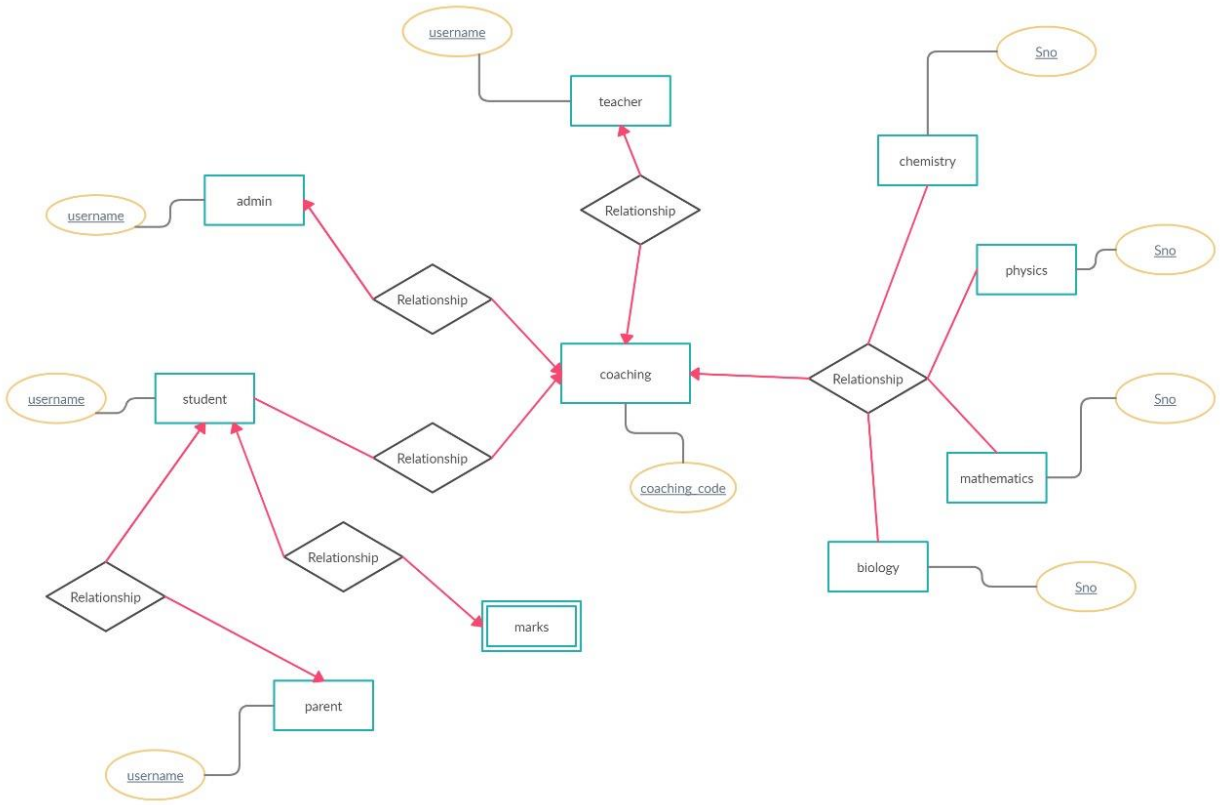
Teacher(**username**,password,first_name,last_name,address,contact,subject,gender)

Weak Entities:

Marks(physics,chemistry,mathematics,biology,)

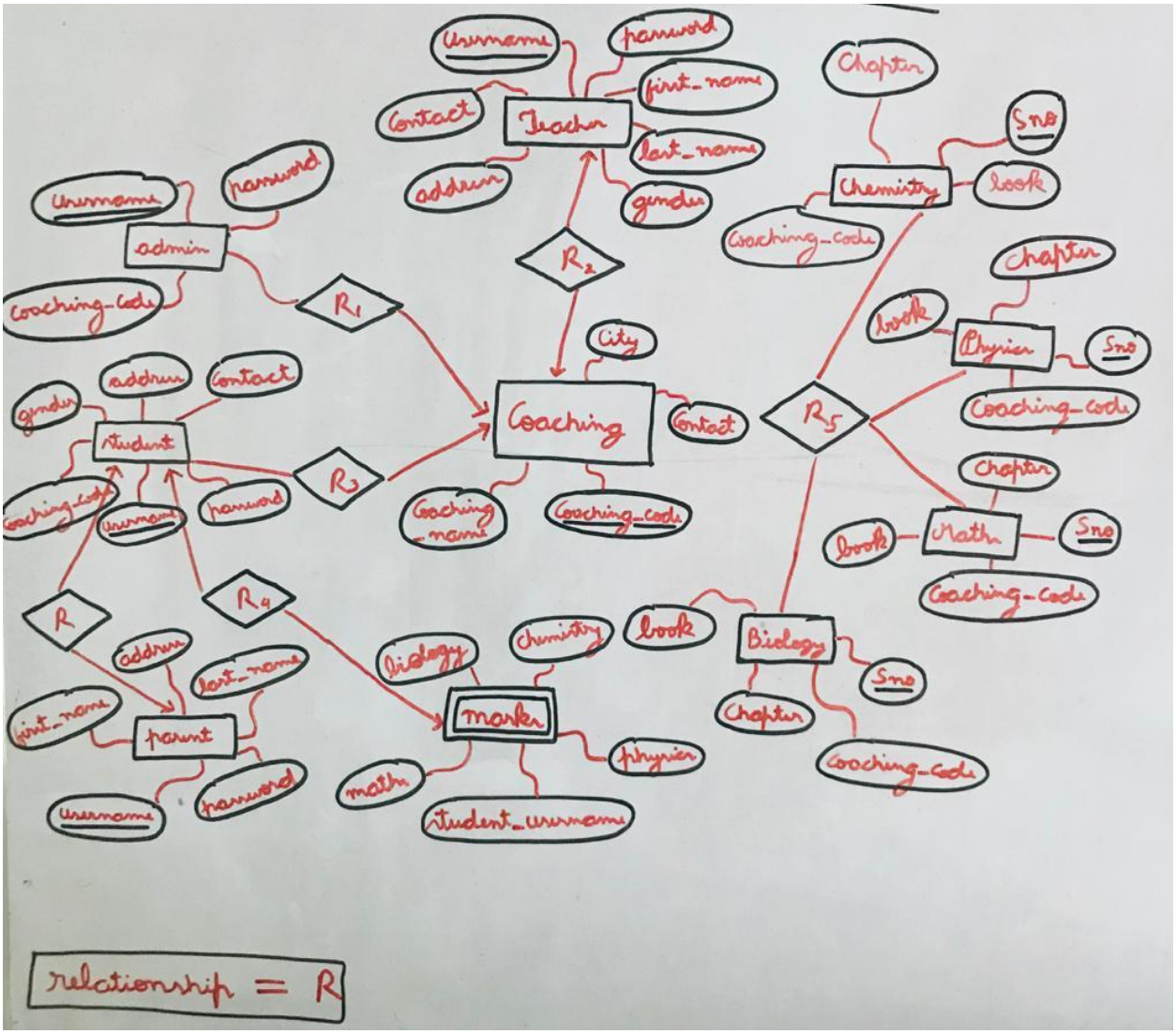
ER Diagram (Modified)

Only with Primary Keys



ERD with all attributes mentioned

NB – We could not add all the attributes in the online er diagram tool as it required paid subscription



Schemas (Modified)

Coaching(**coaching_code**,coaching_name,city,address,contact)

Admin(**username**,password,coaching_code)

Biology(**Sno**,chapter,book,coaching_code)

Mathematics(**Sno**,chapter,book,coaching_code)

Physics(**Sno**,chapter,book,coaching_code)

Chemistry(**Sno**,chapter,book,coaching_code)

Student(**username**,password,coaching_code,gender,address,first_name,last_name,DOB,contact)

Parent(**username**,password,first_name,last_name,address,contact,student_username)

Teacher(**username**,password,first_name,last_name,address,contact,subject,salary,gender,coaching_code)

Phase 2

1.Table for normalization

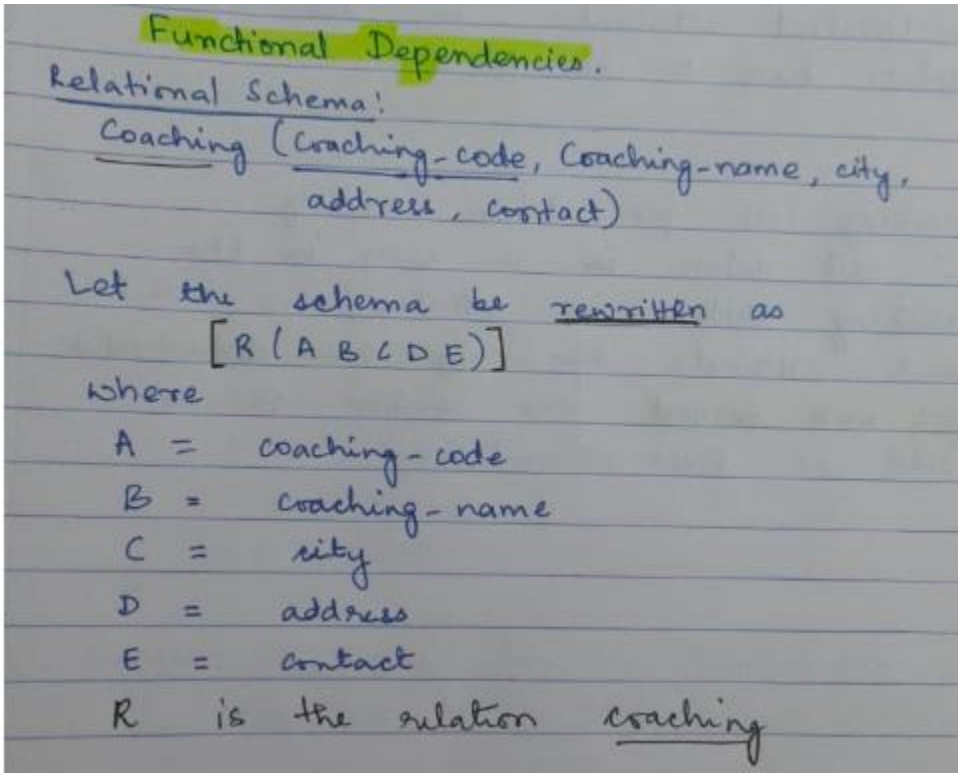
Schema chosen : COACHING

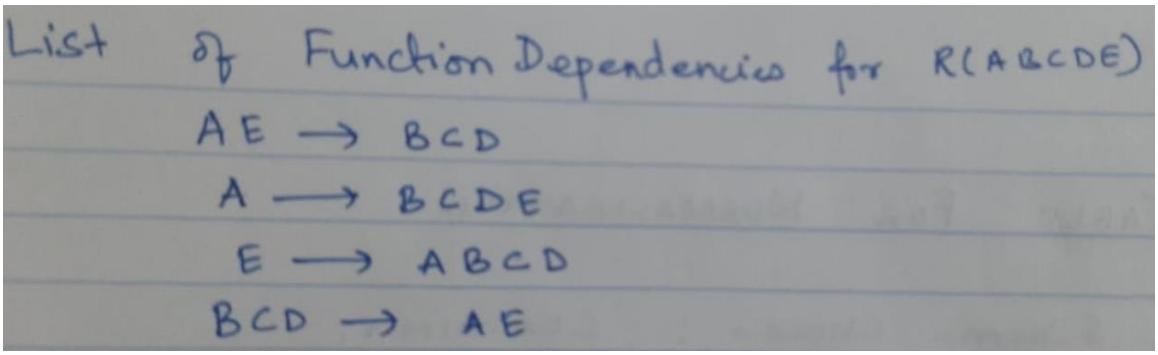
Coaching(Coaching_code, Coaching_name, city, address, contact)

Table(with records):

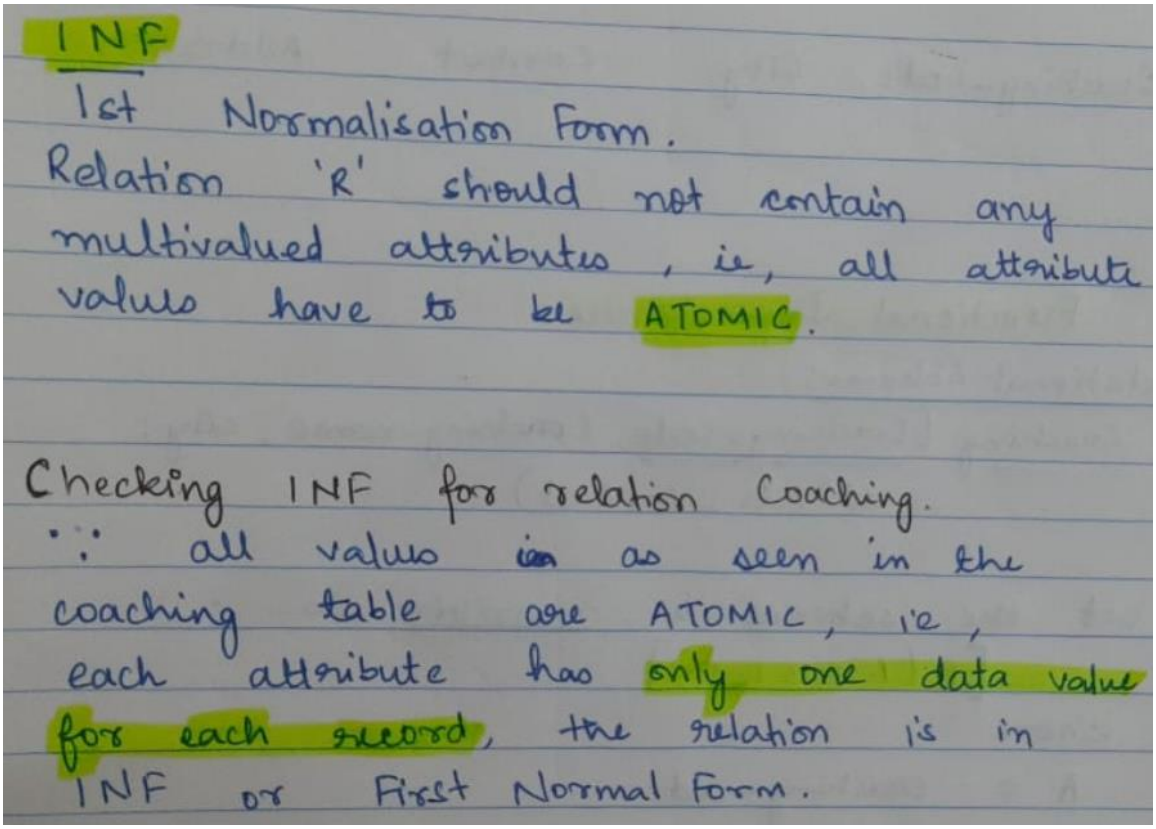
coaching_code	coaching_name	city	address	contact
C1001	Prerna Classes	Jamshedpur	Prerna Towers,Bistupur,831001	7541097627
C1002	Prerna Classes	Ranchi	New Town Aggarwal Apartment,Airport Road,831002	7541097621
C1003	Allen Coaching Classes	Kota	Planetoriam Road, Birla Temple,731001	7541097622
C1004	FIIT-JEE Coaching Classes	Kota	M G Road,South End,731002	7541097623
C1005	Allen Coaching Classes	Kota	M G Road,South End,731002	7541097624
C1006	Allen Coaching Classes	Kota	M G Road,North End,731002	7541097625

2.Functional Dependencies

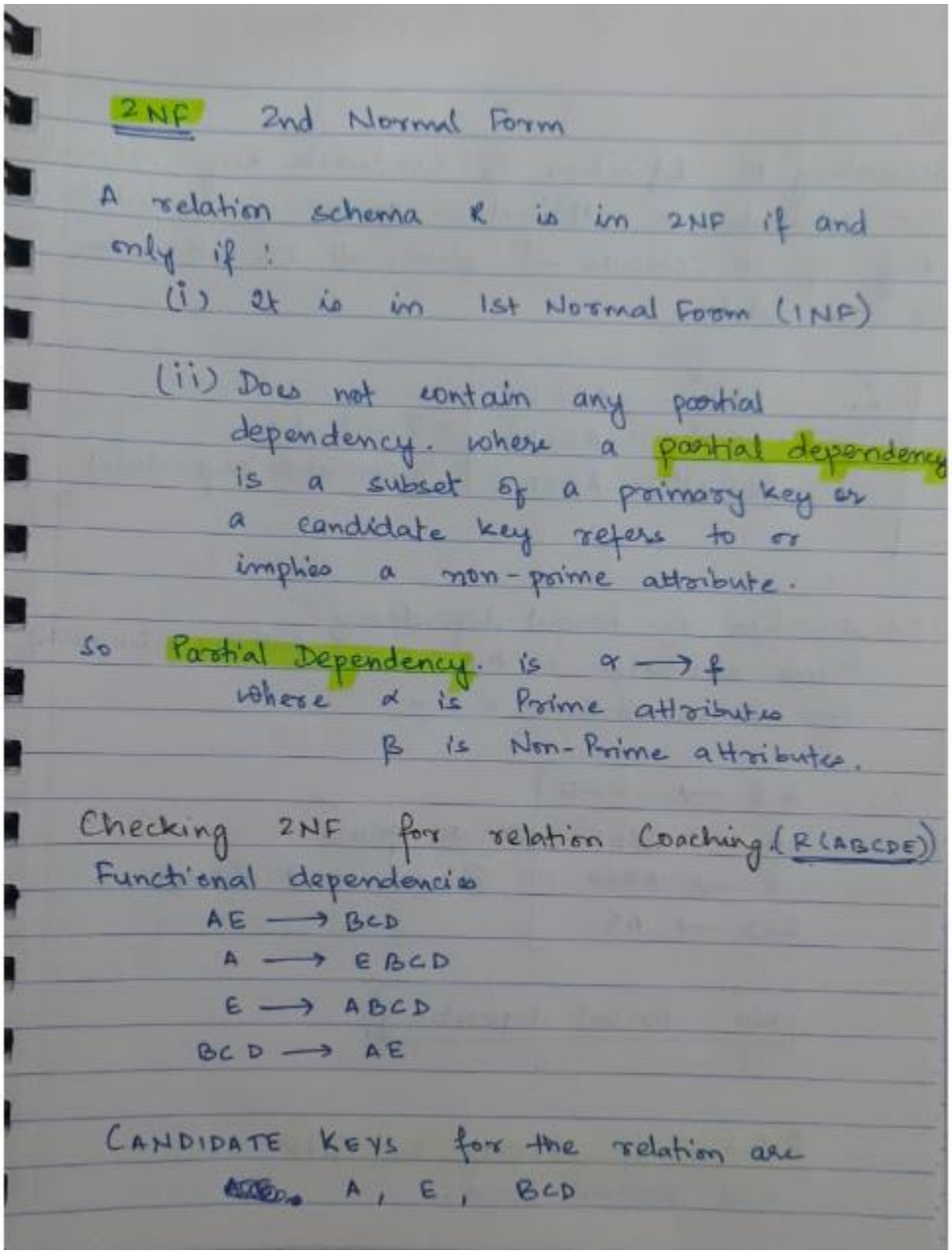




3.First Normal Form(1NF)



4.Second Normal Form(2NF)



2NF 2nd Normal Form

A relation schema R is in 2NF if and only if :

(i) It is in 1st Normal Form (1NF)

(ii) Does not contain any partial dependency. where a partial dependency is a subset of a primary key or a candidate key refers to or implies a non-prime attribute.

So Partial Dependency. is $\alpha \longrightarrow \beta$ where α is Prime attributes β is Non-Prime attributes.

Checking 2NF for relation Coaching.(R(ABCDE))

Functional dependencies

AE $\longrightarrow$ BCD

A $\longrightarrow$ EBCD

E $\longrightarrow$ ABCD

BCD $\longrightarrow$ AE

CANDIDATE KEYS for the relation are

A, E, BCD

As,

According to definition of Candidate key, minimal an attribute or attributes can be a candidate key if its closure set gives all the attributes in the relation.

$\therefore$ $(A)^+ = ABCDE$
$(E)^+ = ABCDE$
$(BCD)^+ = ABCDE$
$\Rightarrow$ Justification of candidate keys selected.

Identification of Partial dependency.

Prime attributes = A B C D E (attributes in all CK)

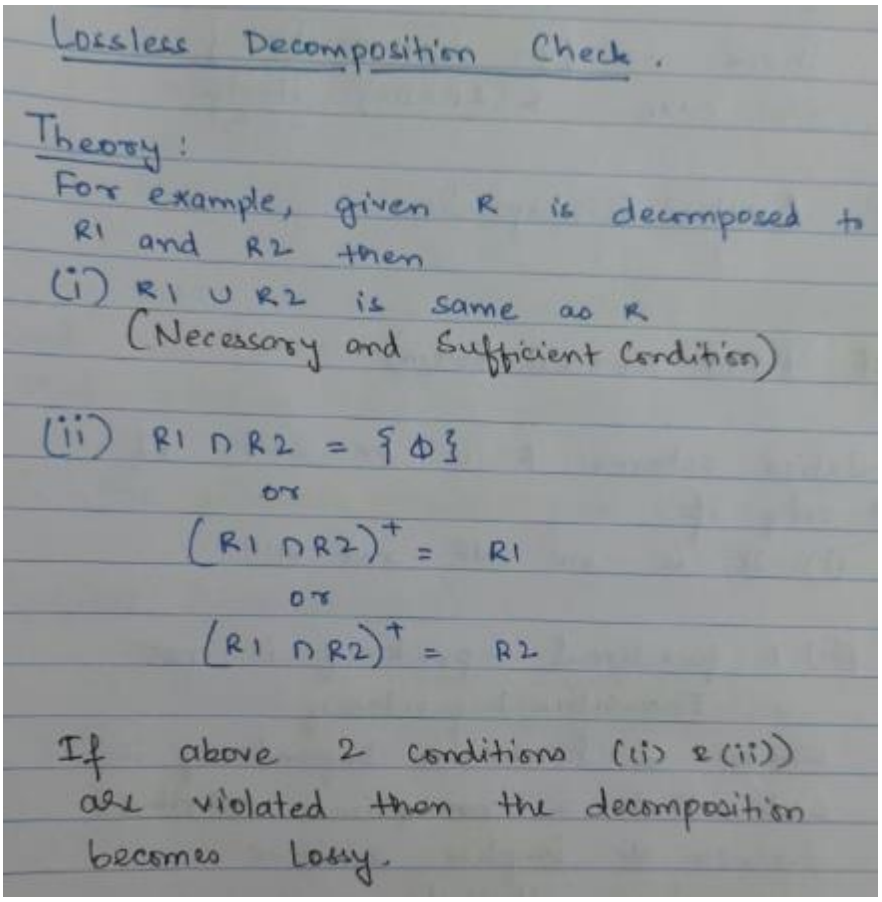
Non-Prime attributes = –

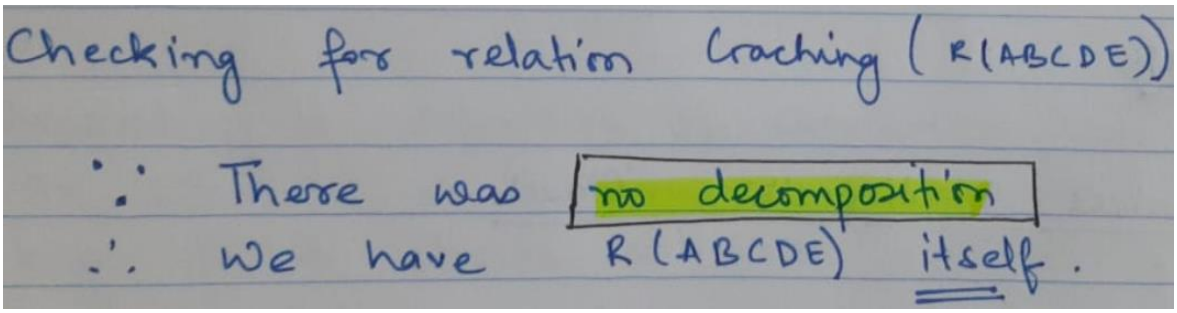
$\therefore$ $AE \longrightarrow BCD$
$A \longrightarrow BCDE$
$E \longrightarrow ABCD$
$BCD \longrightarrow AE$
$\Rightarrow$ all follow Prime $\longrightarrow$ Prime

$\therefore$ No Partial dependancy

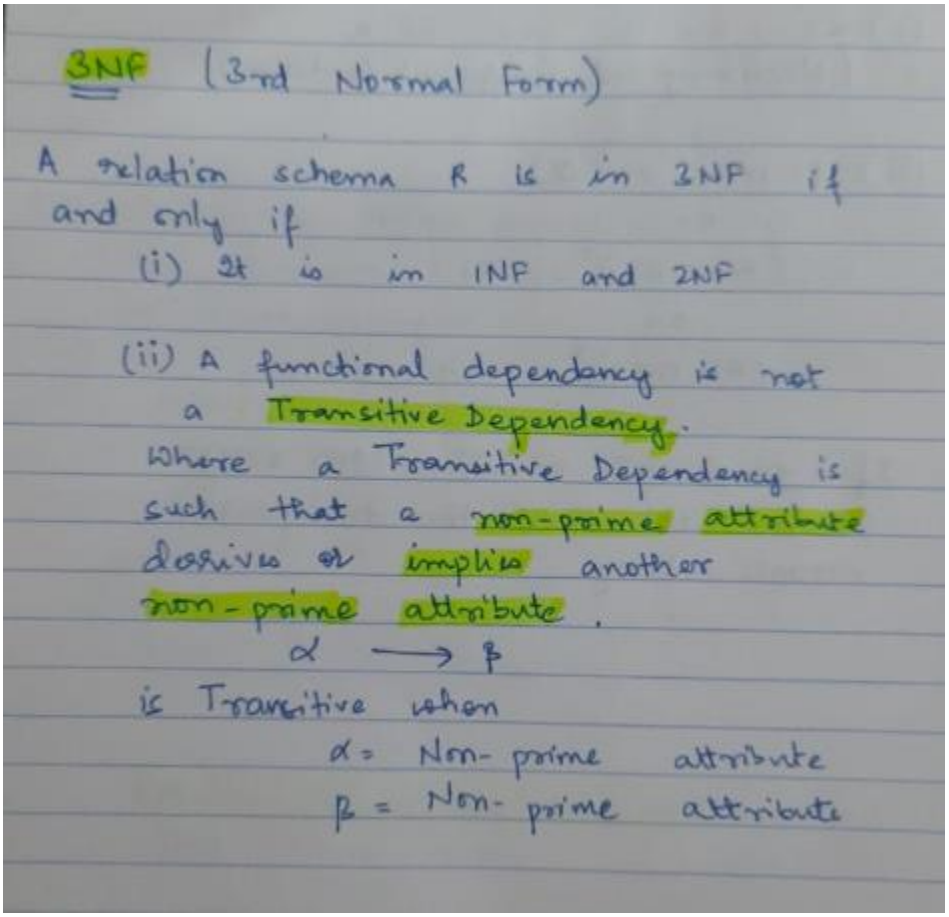
$\therefore$ Relation is already in 2NF or 2nd Normal Form.

5.Decomposition





6.Third Normal Form(3NF)



3NF (3rd Normal Form)

A relation schema R is in 3NF if and only if

(i) It is in 1NF and 2NF

(ii) A functional dependency is not a Transitive Dependency.

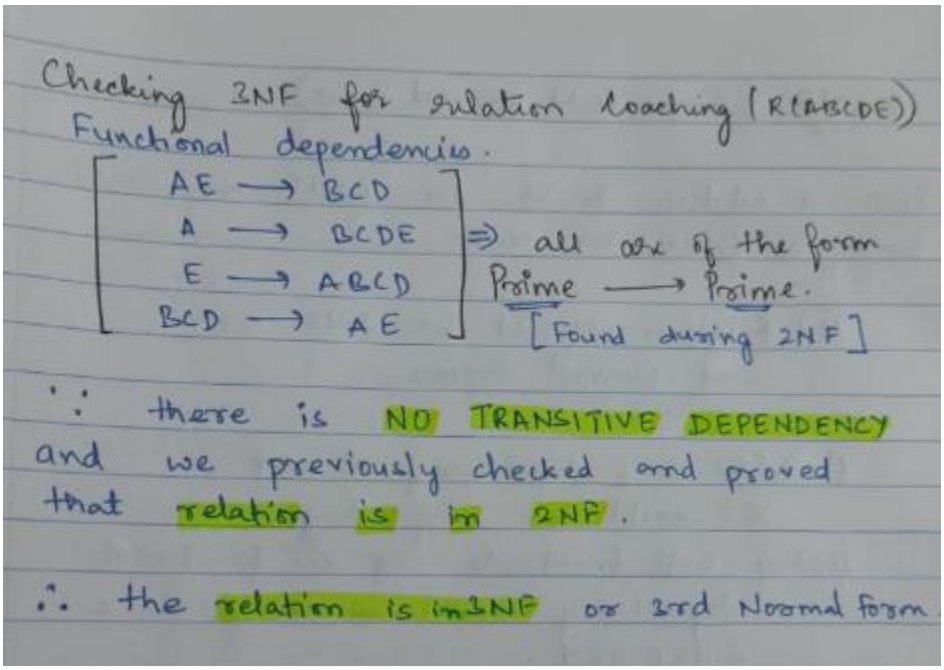
Where a Transitive Dependency is such that a non-prime attribute derives or implies another non-prime attribute.

$\alpha \longrightarrow \beta$

is Transitive when

α = Non-prime attribute

β = Non-prime attribute



Checking 3NF for relation Coaching (R(ABCDE))

Functional dependencies.

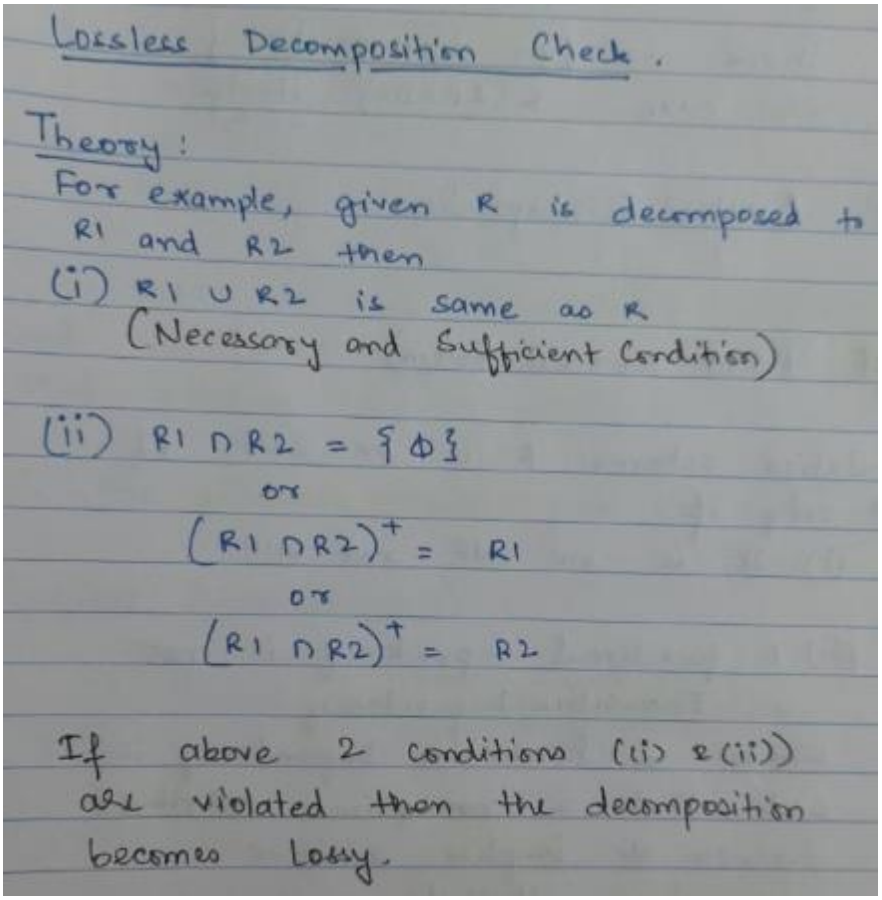
AE → BCD
A → BCDE
E → ABCD
BCD → AE

⇒ all are of the form Prime → Prime. [Found during 2NF]

∵ there is NO TRANSITIVE DEPENDENCY and we previously checked and proved that relation is in 2NF.

∴ the relation is in 3NF or 3rd Normal form.

7.Decomposition



Lossless Decomposition Check.

Theory:

For example, given R is decomposed to R1 and R2 then

(i) R1 ∪ R2 is same as R
(Necessary and Sufficient Condition)

(ii) $R1 \cap R2 = \{\Phi\}$

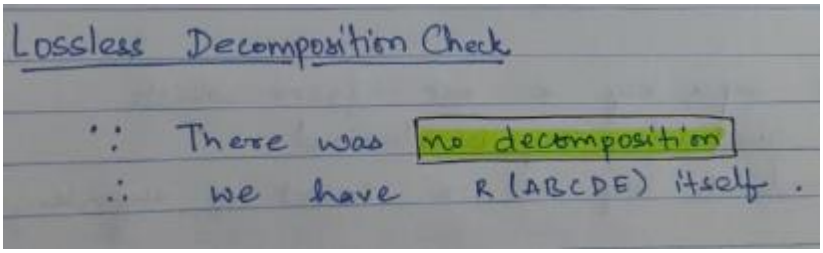
or

$(R1 \cap R2)^+ = R1$

or

$(R1 \cap R2)^+ = R2$

If above 2 conditions (i) & (ii)) are violated then the decomposition becomes Lossy.

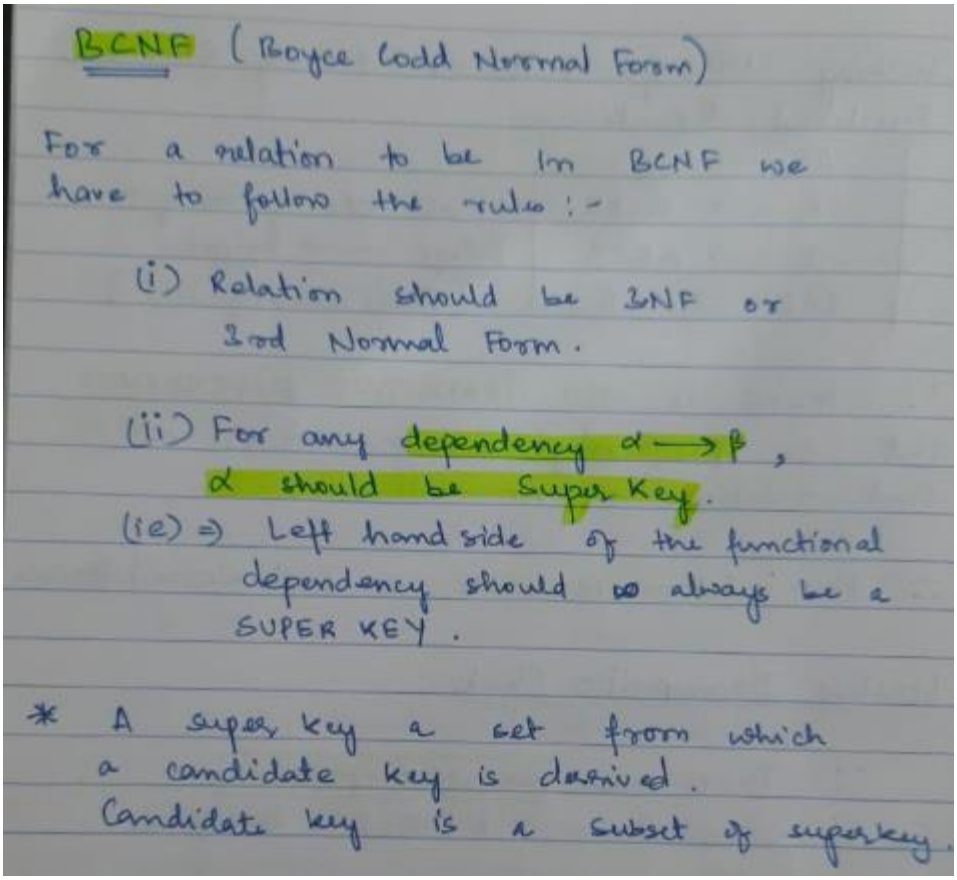


Lossless Decomposition Check

∵ There was no decomposition

∴ we have R(ABCDE) itself.

8.Boyce Codd Normal Form(BCNF)



BCNF (Boyce Codd Normal Form)

For a relation to be in BCNF we have to follow the rules :-

(i) Relation should be 3NF or 3rd Normal Form.

(ii) For any dependency $\alpha \longrightarrow \beta$, α should be Super Key.

(ie) => Left hand side of the functional dependency should be always be a SUPER KEY.

* A super key a set from which a candidate key is derived.
Candidate key is a subset of superkey.

Checking BCNF for relation coaching (R(ABCDE))

Functional dependencies.

$AE \longrightarrow BCD$

$A \longrightarrow BCDE$

$E \longrightarrow ABCD$

$BCD \longrightarrow AE$

R(ABCDE) was already in 3NF.

Candidate keys = A, E, BCD.

(calculated and justified previously)

$\therefore$ $AE \longrightarrow BCD$, $A \longrightarrow BCDE$, $E \longrightarrow ABCD$, $BCD \longrightarrow AE$ $\Rightarrow$ all are of the form $\alpha \longrightarrow \beta$ where $\alpha \in$ {superkey}

$\therefore$ they are in BCNF.

* AE is a superkey as

$(AE)^+ = ABCDE$.

A is a superkey as

$(A)^+ = ABCDE$

E is a superkey as

$(E)^+ = ABCDE$

BCD is a superkey as

$(BCD)^+ = ABCDE$

$\therefore$ R(ABCDE) is BCNF form.

$\therefore$ The relation R(ABCDE)

(ie)

Coaching (coaching-code, coaching-name, city, address, contact)

is already decomposed to BCNF or Boyce Codd Normal Form.

9. Decomposition

Since the relation was already in Boyce Codd Normal Form, no decomposition is required.

Final Schemas:

Coaching(**coaching_code**,coaching_name,city,address,contact)

Admin(**username**,password,coaching_code)

Biology(**Sno**,chapter,book,coaching_code)

Mathematics(**Sno**,chapter,book,coaching_code)

Physics(**Sno**,chapter,book,coaching_code)

Chemistry(**Sno**,chapter,book,coaching_code)

Student(**username**,password,coaching_code,gender,address,first_name,last_name,DOB,contact)

Parent(**username**,password,first_name,last_name,address,contact,student_username)

Teacher(**username**,password,first_name,last_name,address,contact,subject,salary,gender,coaching_code)

Phase 3

Hardware Requirement :

RAM : 1 GB or more

Memory : 512 MB or more

Processor : 2.5 GHz or more

Software Requirements :

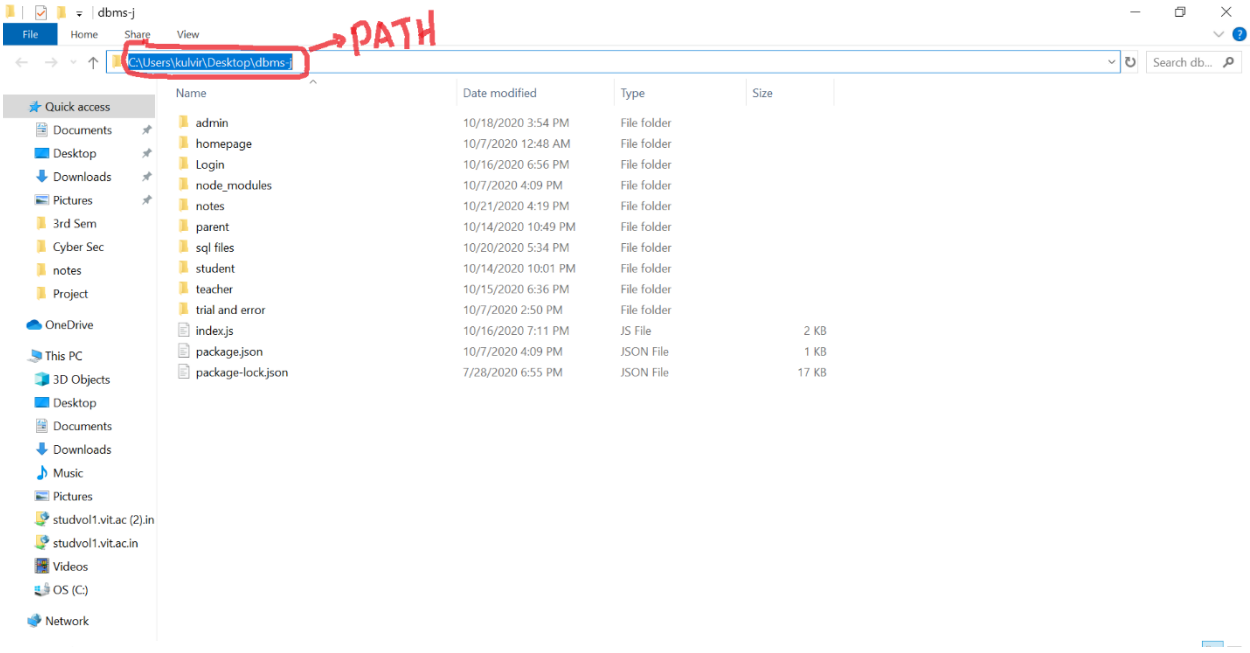
Nodejs https://nodejs.org/en/

MySQL https://dev.mysql.com/downloads/mysql/

How to use the application(Help File):

Step 1:

Install the required software on the computer which has met the hardware requirements. Then download/copy the entire project in your system. Copy the path of the project folder. The path can be found on the top of the folder window as shown below



Step 2:

Now open the command prompt terminal by pressing Ctrl+R, then type in cmd and hit enter.

Write the command chdir, hit spacebar and paste the copied path and then hit enter.

Step 3:

Then simply type nodemon index.js and press enter.

Step 4:

Now go to the link given below and access the website.

http://localhost:3000/homepage

Flow of Control:

Homepage

This is the landing page or the homepage of the website. All the routes of other pages are linked through this page. The page contains the various login buttons for teachers, parents, students and admins along with a welcome message.

Student Login

On clicking the student login button on the homepage, the website is routed to this page. This page consists of a form which asks the user to enter their username and password and a login button. If the user enters the wrong credentials then the control shifts to an error page. If the credentials are correct the control shifts to the user's account.

Teacher Login

On clicking the teacher login button on the homepage, the website is routed to this page. This page consists of a form which asks the user to enter their username and password and a login button. If the user enters the wrong credentials then the control shifts to an error page. If the credentials are correct the control shifts to the user's account.

Parent Login

On clicking the parent login button on the homepage, the website is routed to this page. This page consists of a form which asks the user to enter their username and password and a login button. If the user enters the wrong credentials then the control shifts to an error page. If the credentials are correct the control shifts to the user's account.

Admin Login

On clicking the admin login button on the homepage, the website is routed to this page. This page consists of a form which asks the user to enter their username and password and a login button. If the user enters the wrong credentials then the control shifts to an error page. If the credentials are correct the control shifts to the user's account. Also, a register button is given for a new user to register to the website.

Student Account

This is the student user's account landing page where all the options or features reserved for the user only are displayed in the form of multiple buttons. This page can only be reached after entering the correct credentials in the login page. buttons on this page process the post request and display the output accurately, i.e., whatever the functionality written on the button, that is executed successfully. For ex: on clicking check your details button will process the request and display the detail by fetching the required attributes from the database and project them on the webpage with proper styles applied to it.

Parent Account

This is the parent user's account landing page where all the options or features reserved for the user only are displayed in the form of multiple buttons. This page can only be reached after entering the correct credentials in the login page. buttons on this page process the post request and display the output accurately, i.e., whatever the functionality written on the button, that is executed successfully. For ex: on clicking check your details button will process the request and display the detail by fetching the required attributes from the database and project them on the webpage with proper styles applied to it.

Teacher Account

This is the teacher user's account landing page where all the options or features reserved for the user only are displayed in the form of multiple buttons. This page can only be reached after entering the correct credentials in the login page. buttons on this page process the post request and display the output accurately, i.e., whatever the functionality written on the button, that is executed successfully. For ex: on clicking check your details button will process the request and display the detail by fetching the required attributes from the database and project them on the webpage with proper styles applied to it.

Admin account

This is the admin user's account landing page where a form is displayed which takes text input. The form is styled in such a way that it looks like a sql terminal. The user can enter any sql query and execute it. The run button which is a part of the form is will send the flow of control to the backend which will execute the query and display the result of the query in the form of JSON object.

Forgot Password/Change Password

This page is exclusive for each student, teacher, parent user and can be accessed after logging in to one's account. This page consists of a form where the user is required to enter their new password twice and press the submit to update their account with the new password. The submit buttons shifts the control to the database and updates the new password in the database and replaces the old one.

New Admin Registration

This page is reached when the user clicks the register button on the admin homepage. The webpage displays a form which requires the user to input certain details and contains a submit button which shifts the control to the backend. The code generates a coaching code and uses the city, contact, institute name, address entered by the user via the form to add a new coaching in the database(coaching table). After adding, the backend then adds the username and password and the corresponding coaching code to the database(admin table).

Update marks

This page is reached when the user has logged in the teacher homepage and access the update marks button. This page consists of a form and two buttons. The button view marks sends control to the backend which then displays marks for all students studying in the coaching where the particular teacher is teaching on the webpage. The other button is related to the form. The form requires the

student username whose marks needs to be updated and the new marks. The submit button then taps in the backend which edits the marks from the database(marks table).

Update Reference Books

This page is reached when the user has logged in the teacher homepage and access the update reference books button. This page consists of a form and two buttons. The button view book list sends control to the backend which then displays books for the subject which the teacher teaches in the coaching where the particular teacher is teaching. The other button is related to the form. The form requires the exact topic name as shown in the view book list output whose reference book needs to be updated and the new book name. The submit button then taps in the backend which edits the book from the database.

Check Ward Details

This page is accessed when the user has logged in the parent homepage and access the Check Ward Details button. The button sends the control to the backend and displays the student details by accessing the database and identifying the row to be displayed by using foreign key student_username.

Coaching Details

This page is accessed when the user has logged in the parent, student or teacher homepage and access the coaching details button. The button sends control to the backend and displays the coaching details by accessing the database and identifying the row to be displayed by using foreign key coaching_code.

Check Marks

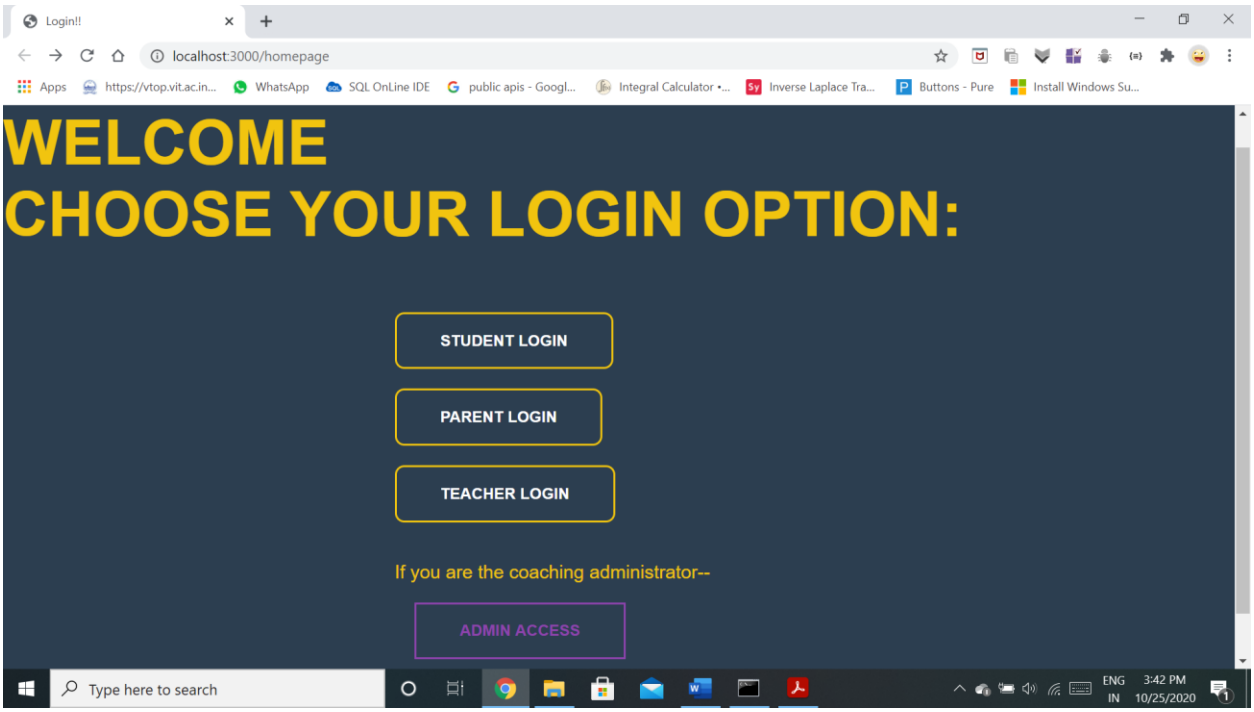
This page is accessed when the user has logged in the parent or student homepage and access the check marks button. The button sends control to the backed and displays the marks of the student by accessing the database and identifying the row to be displayed by using the foreign key student_username.

Tutor Details

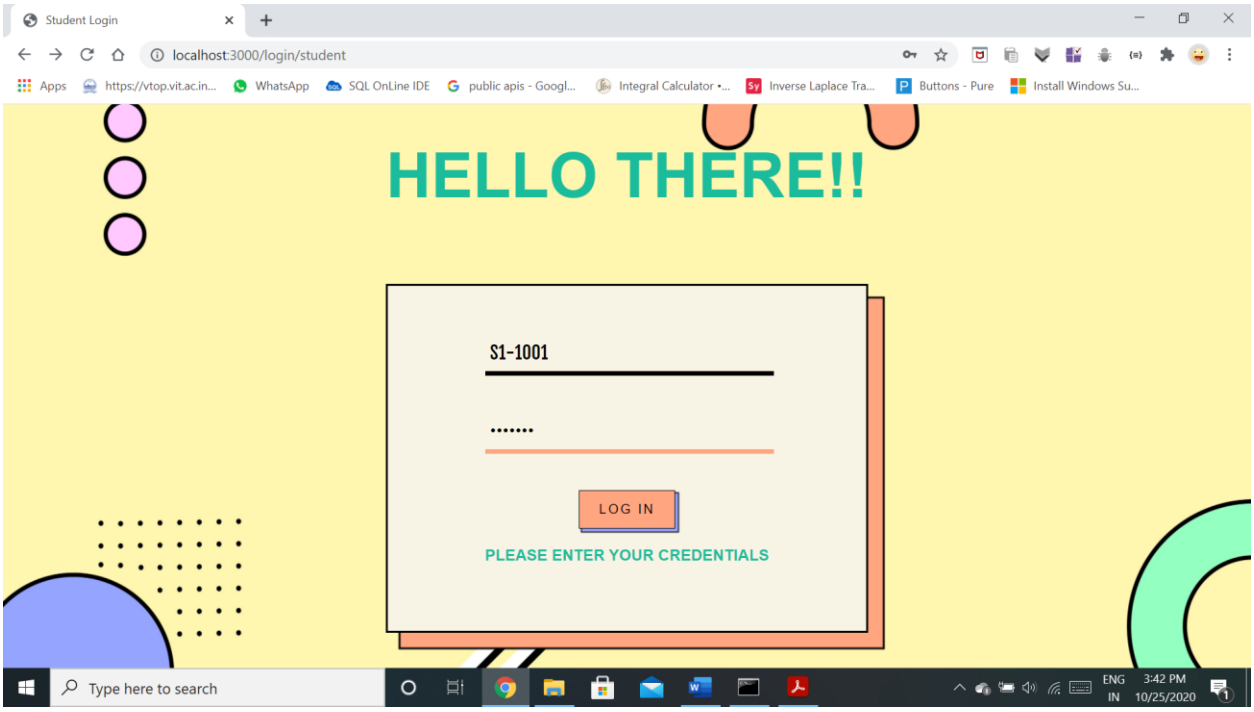
This page is accessed when the user has logged in the parent or student homepage and access the tutor details button. The button sends control to the backed and displays all the teachers' name subject and contact by accessing the database.

Screen Shots

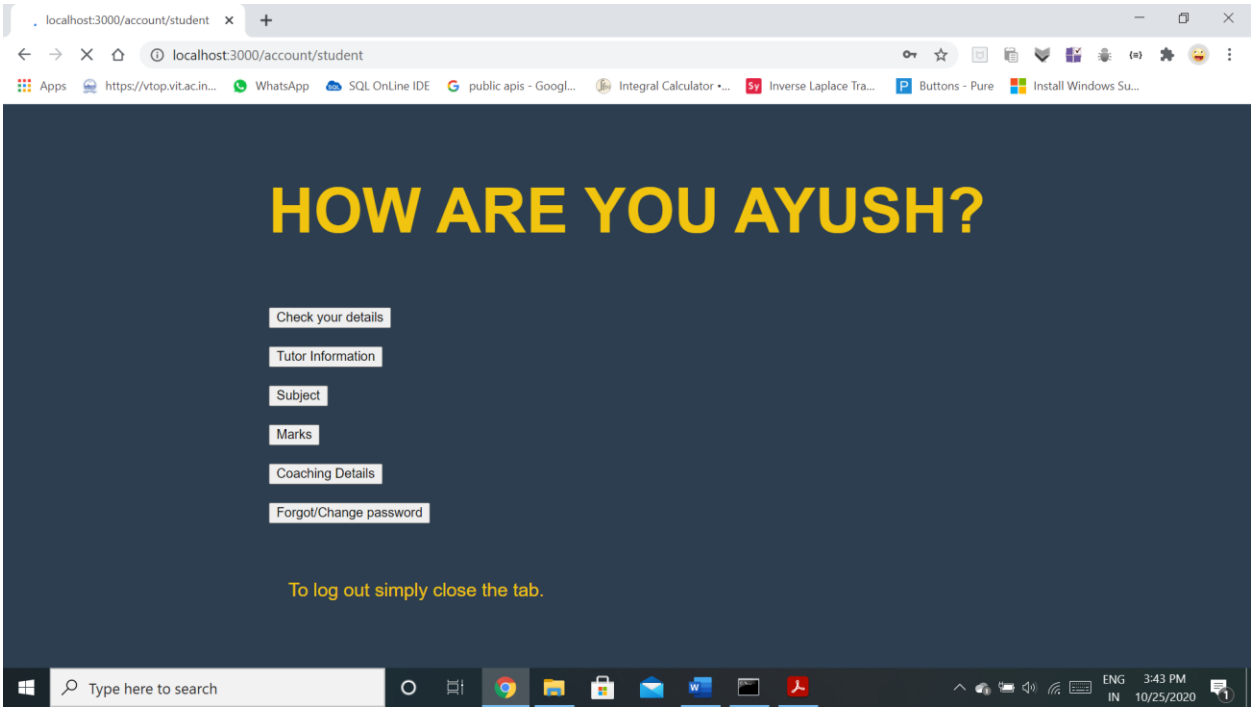
Landing Page:



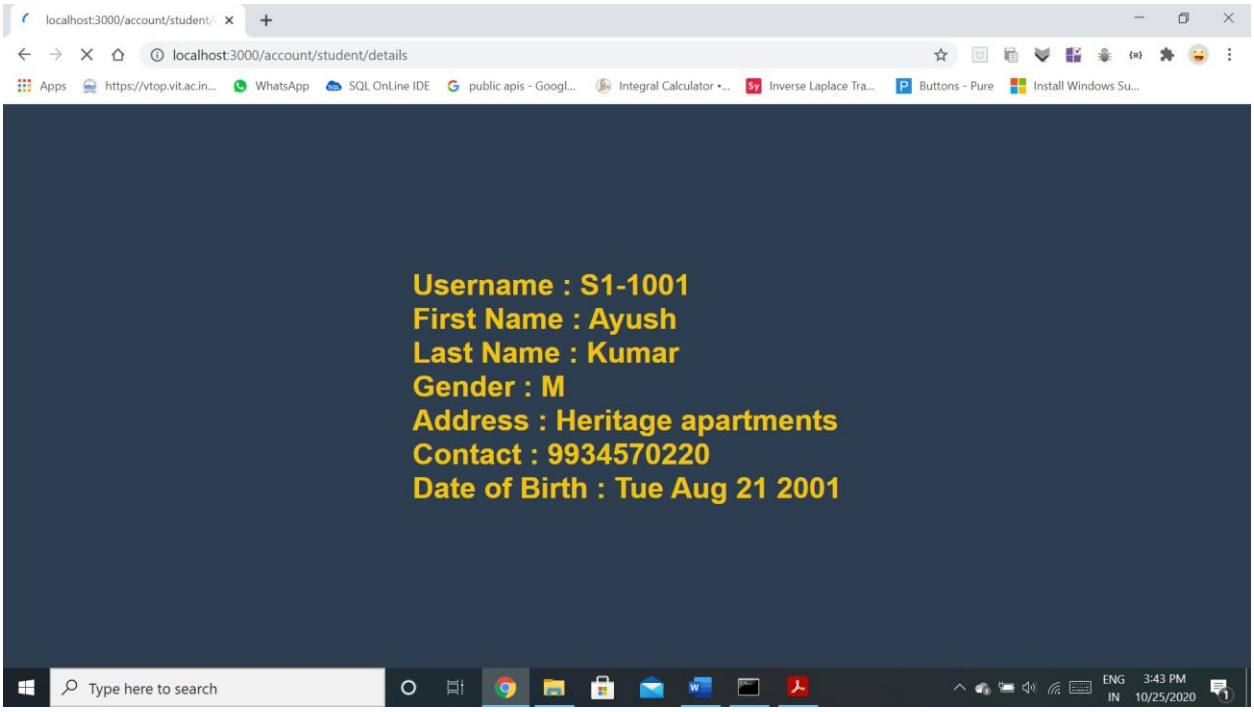
Student Login:



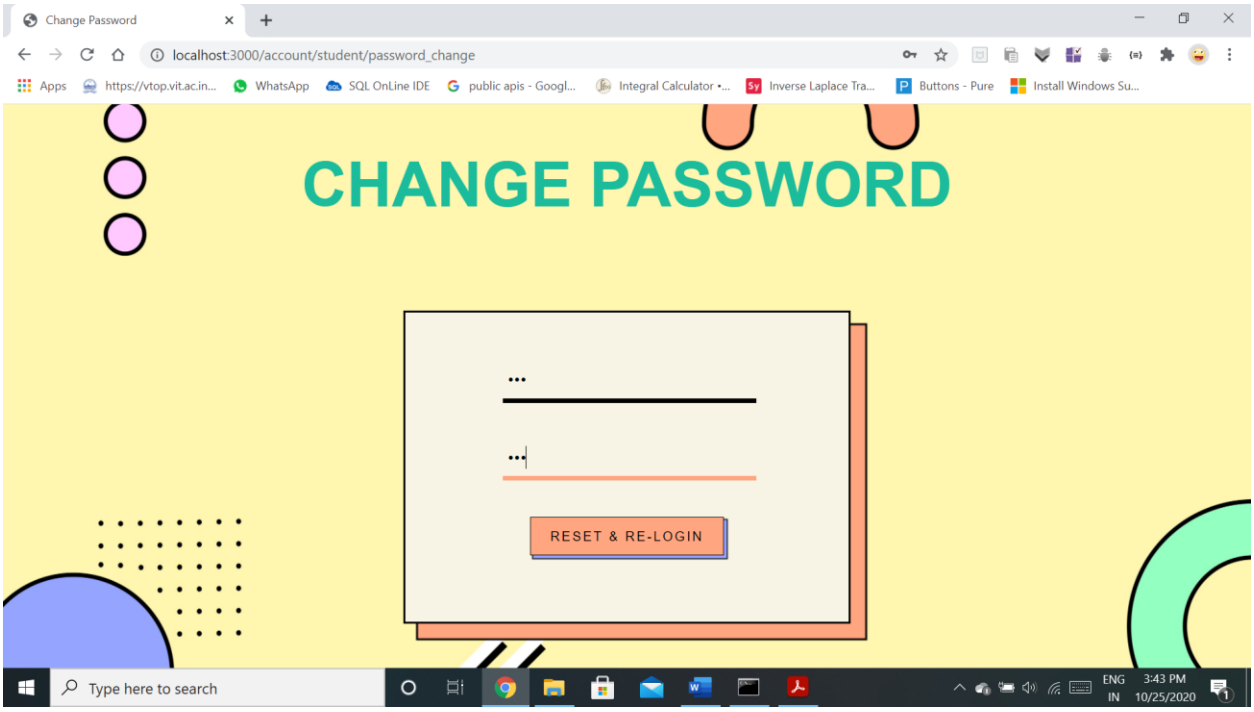
Student Account Homepage:

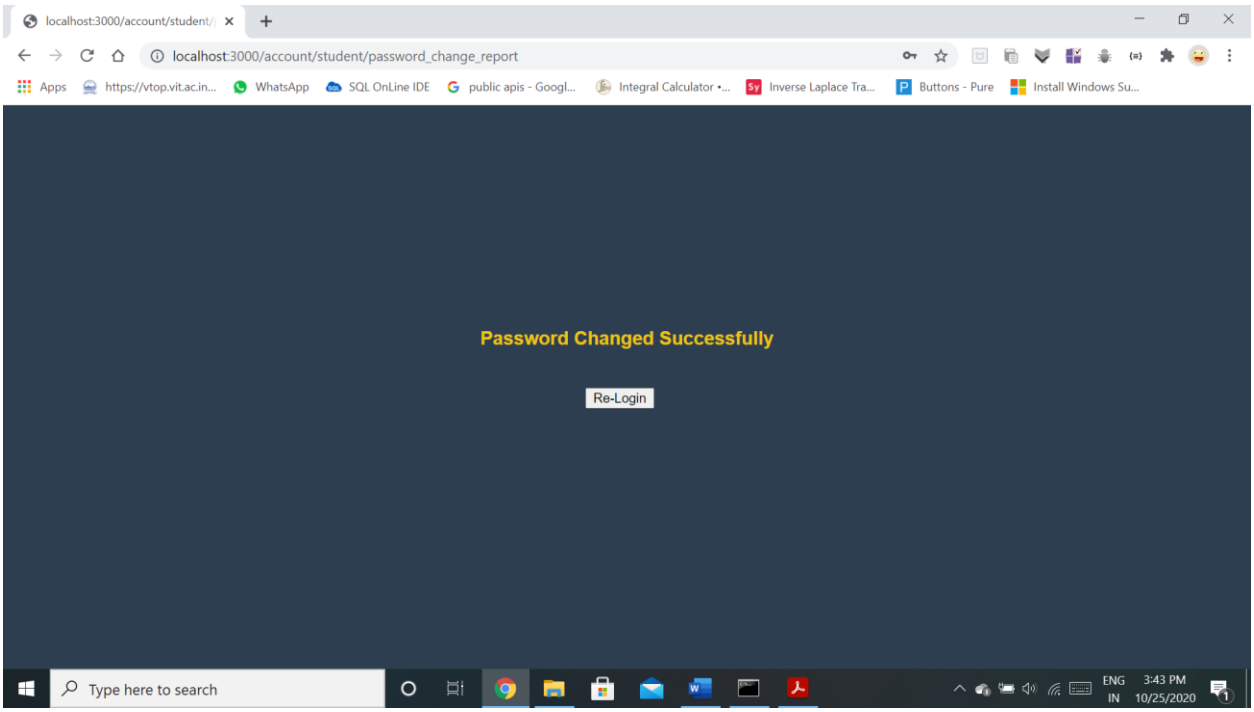


Student Details:

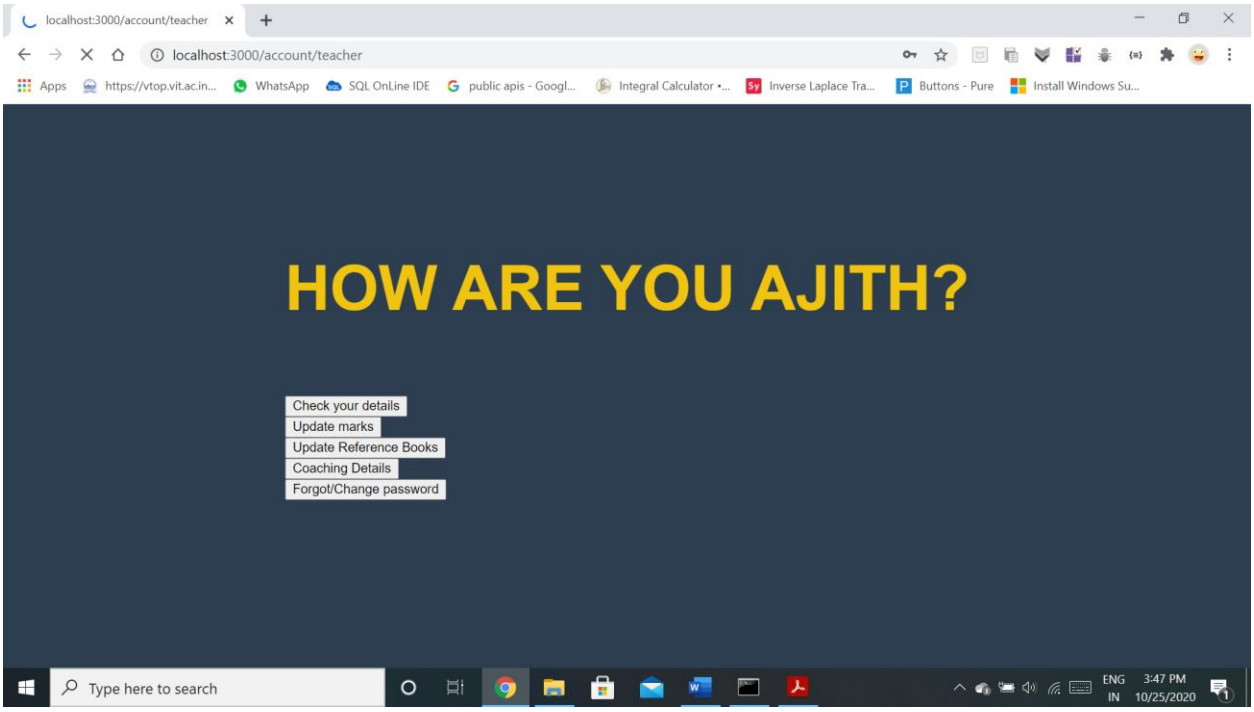


Change/Forgot Password:

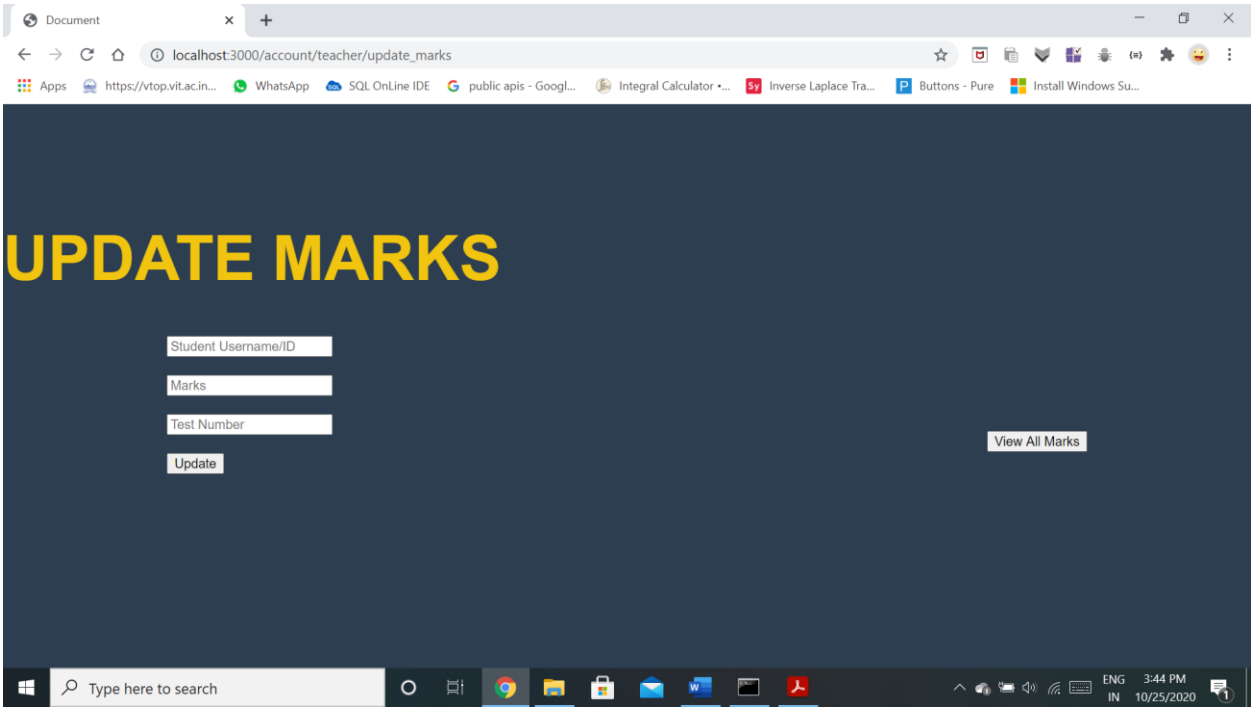




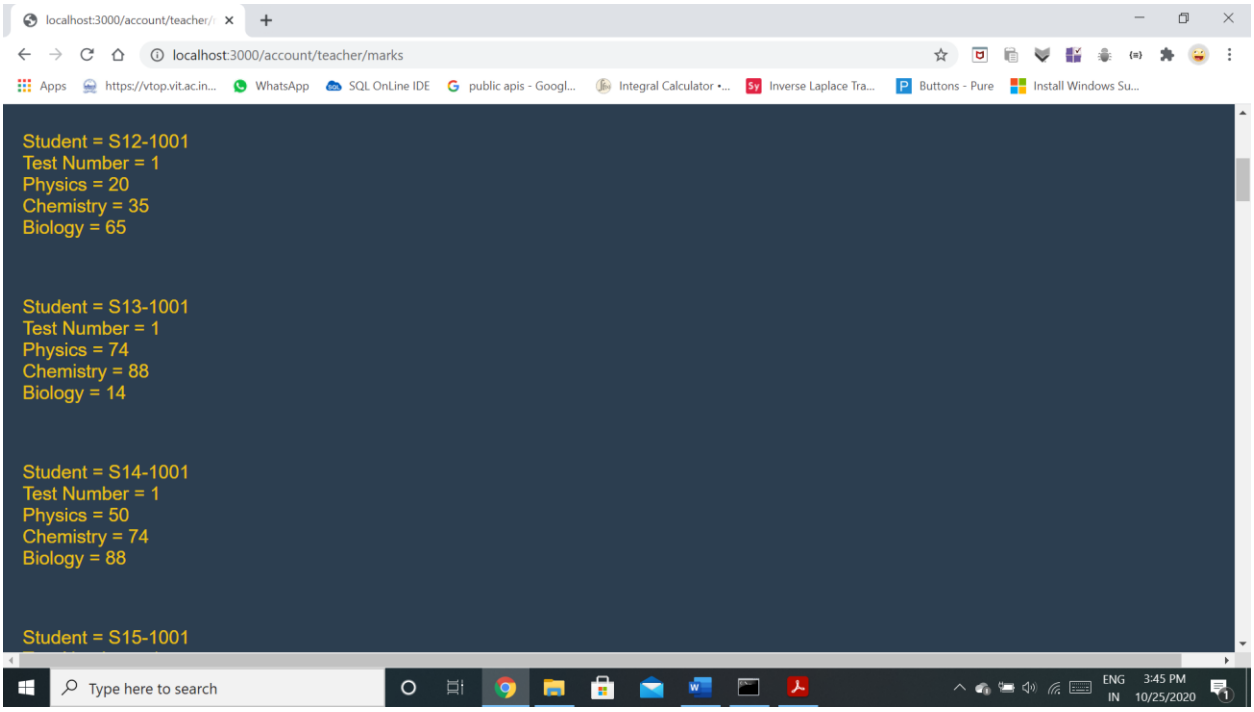
Teacher Account Homepage:



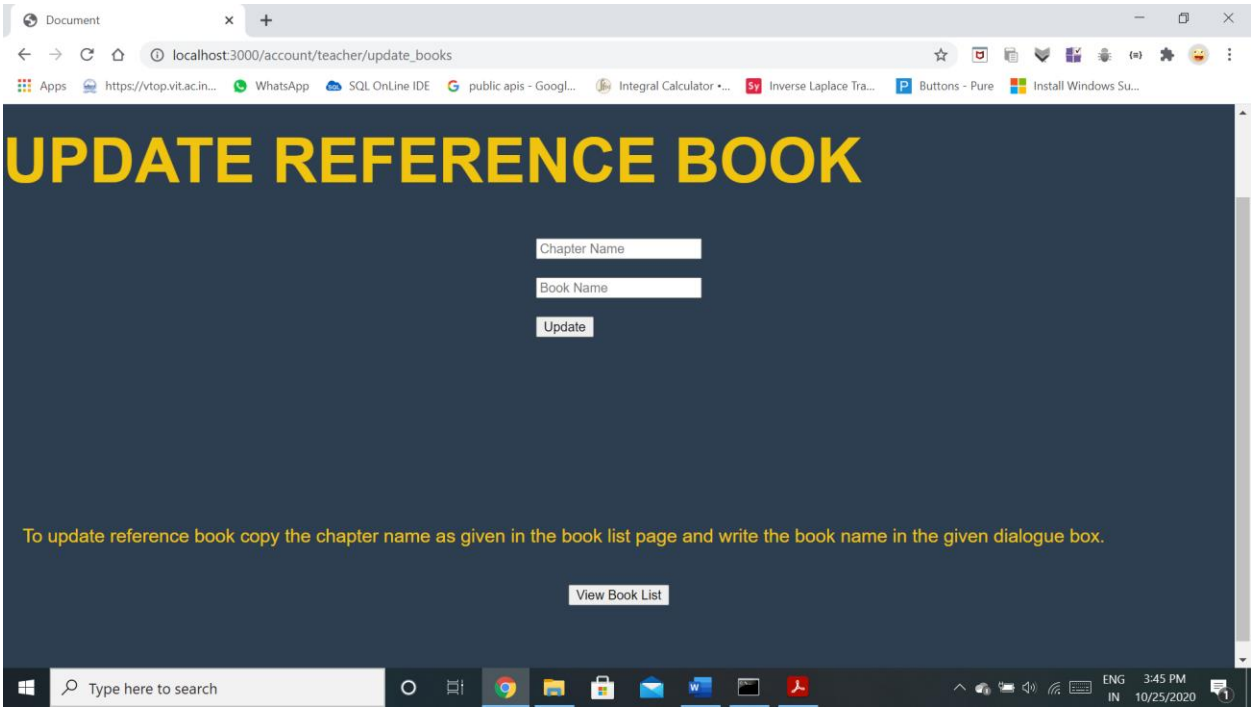
Update Marks (for Teachers):



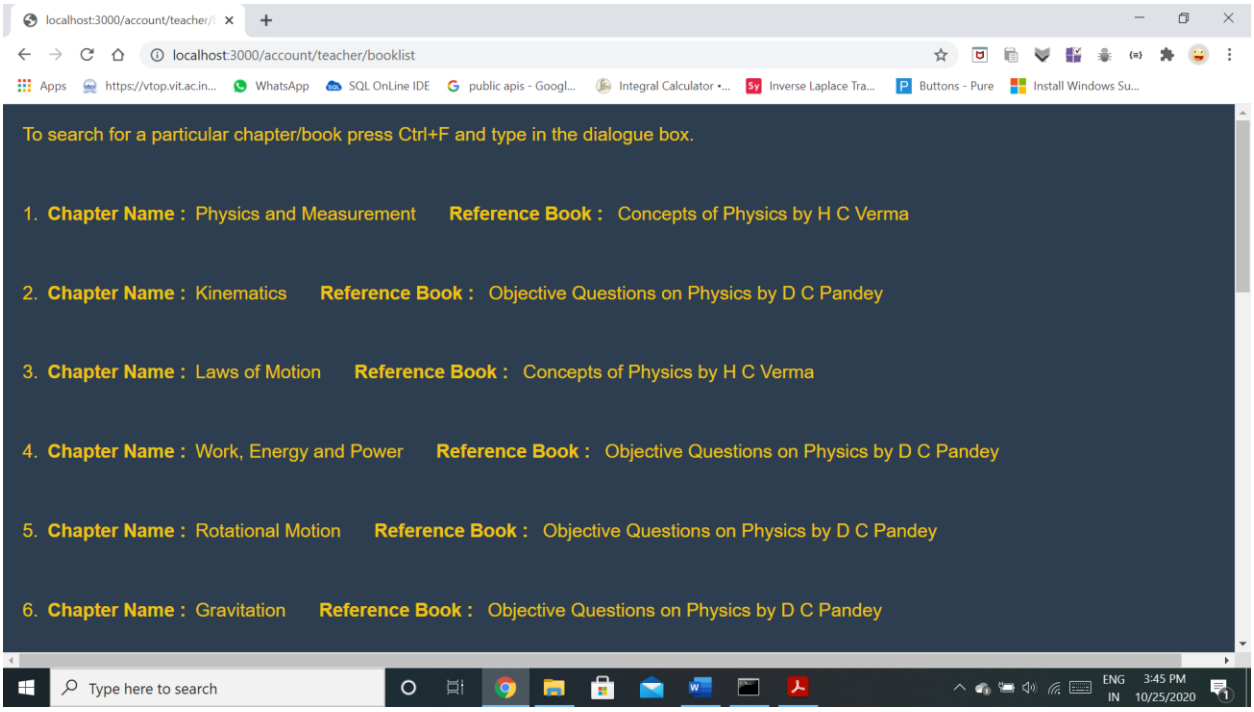
View Marks:



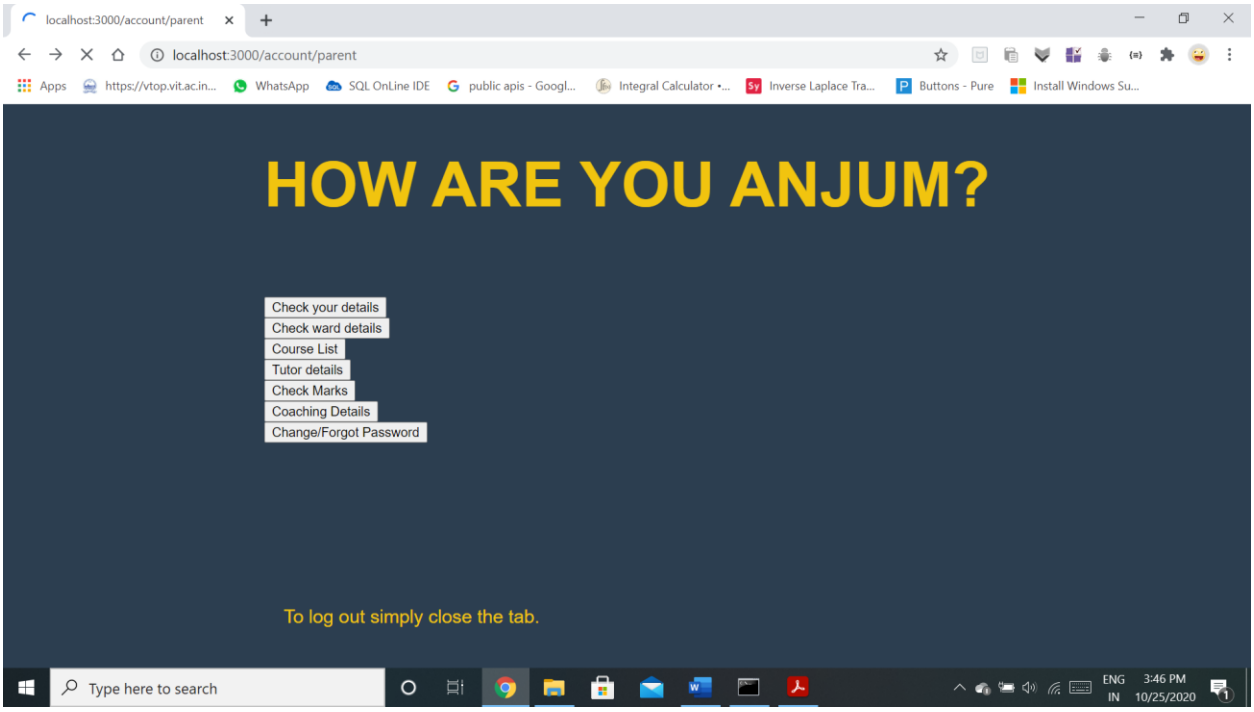
Update Reference Books



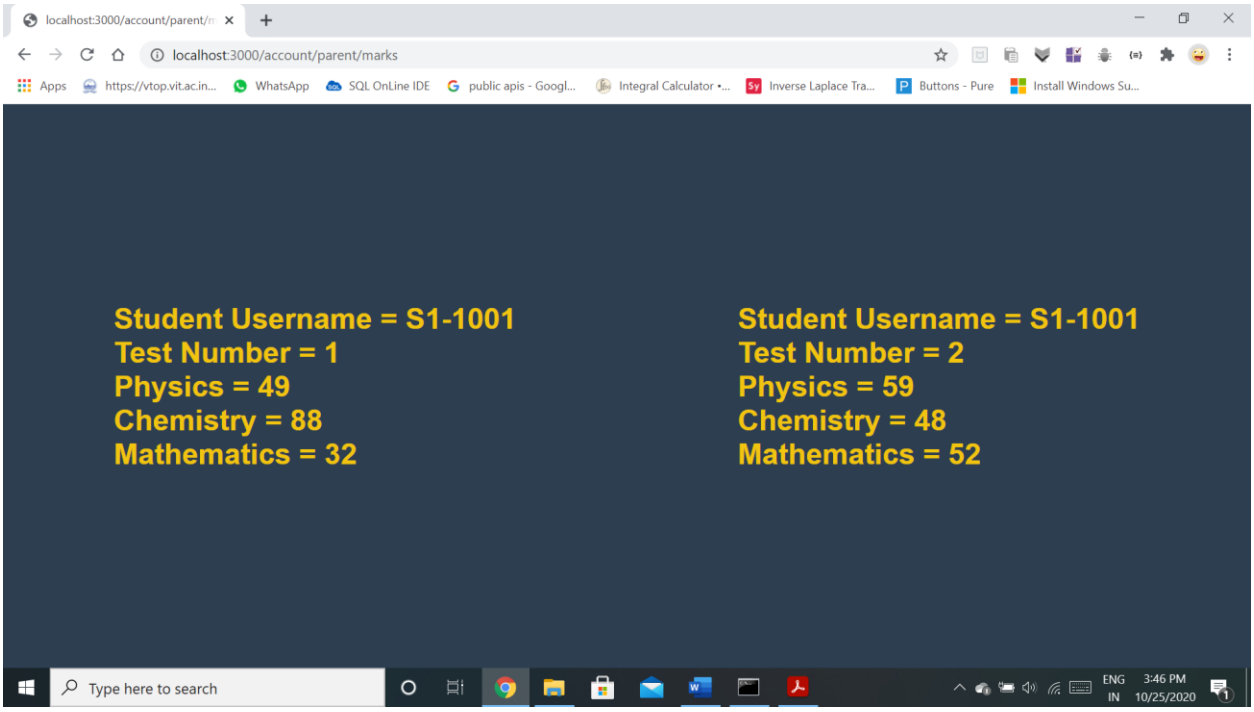
View Booklist:



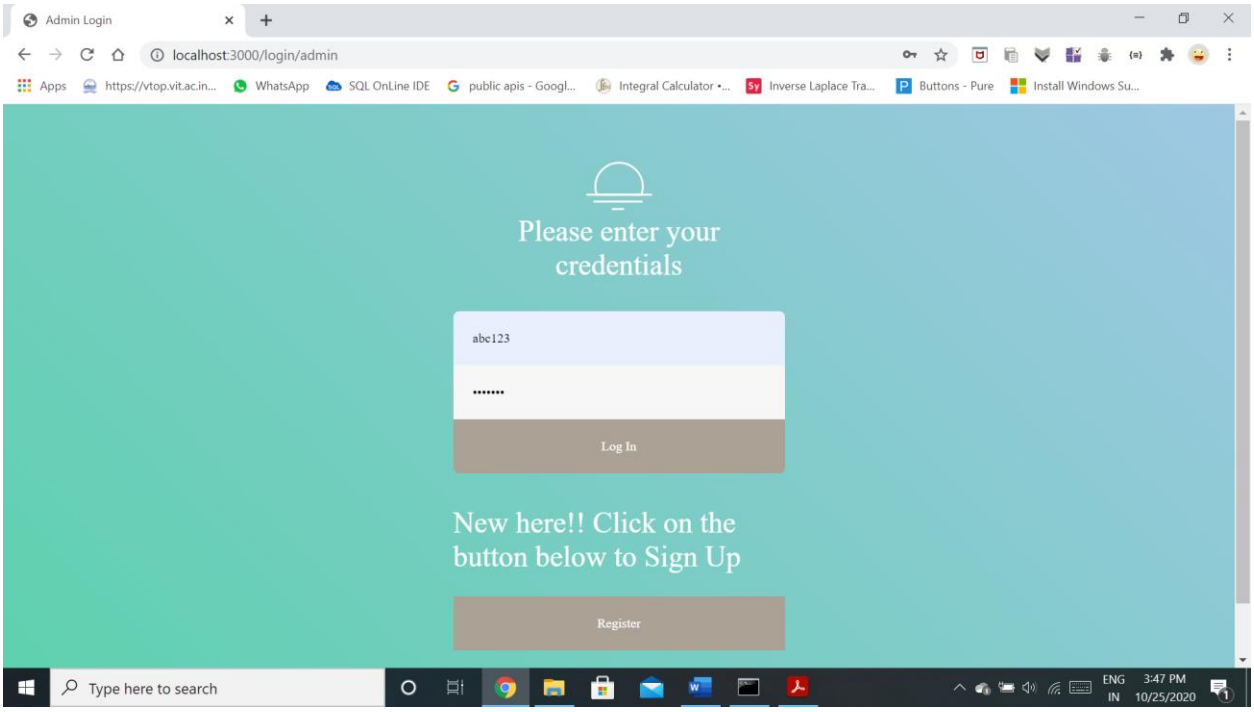
Parent Account Homepage:



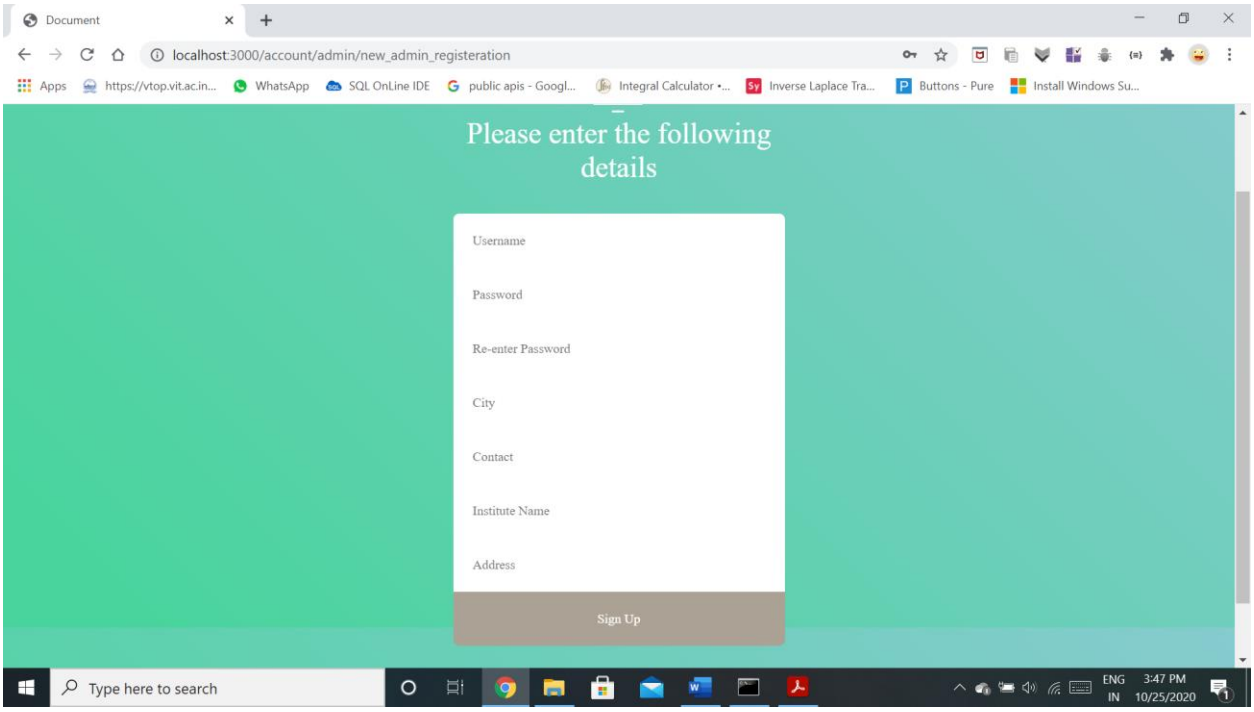
Check Marks (of ward):



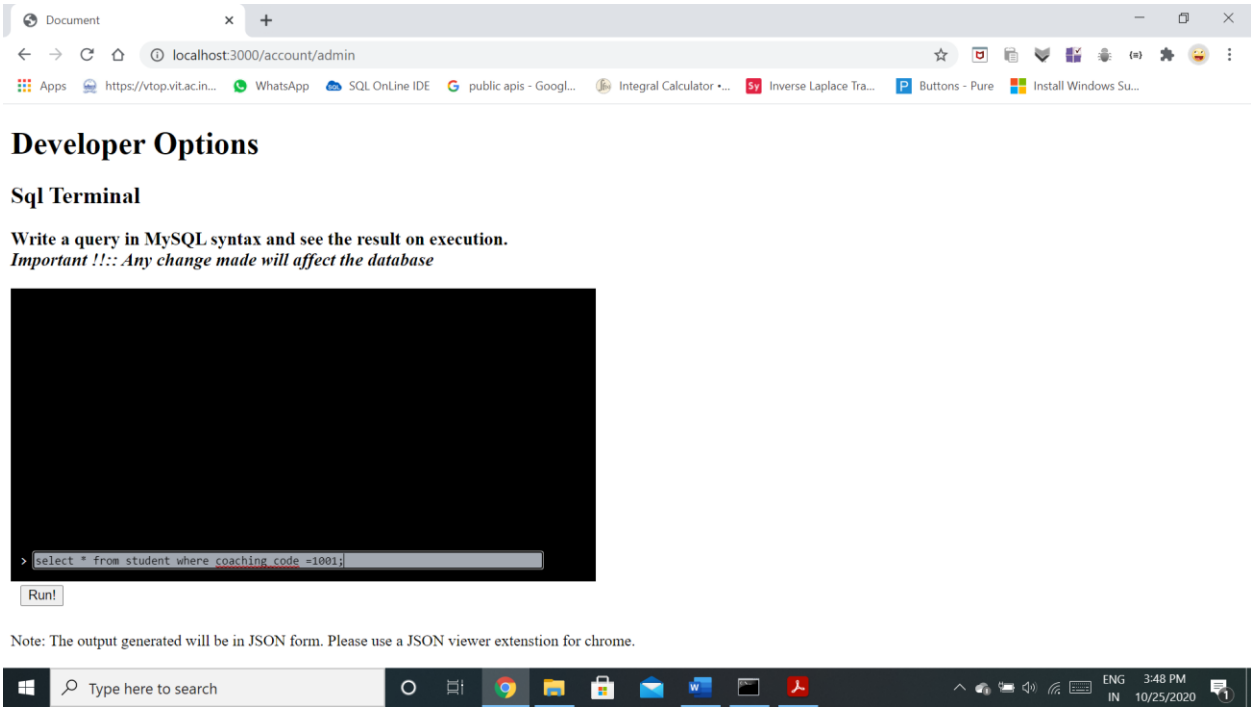
Admin Login Page :



Register/Sign Up Page :



Admin Account Homepage(Sql Terminal like page):



Admin query result page: