

Fundamentals of Database Management Systems (CSE-202)

Bharat कक्षा



Team 76

Anuneet Anand | Himanshi Mathur | Srijan Jain | MD. Talib | Vikas Mahto



Index

CONTENT	Page Number
1. Project Abstract	<u>3-4</u>
2. Week 1 - Stakeholders	<u>5-6</u>
3. Week 2 - Queries	<u>7-9</u>
4. Week 3 - Database Schema	<u>10</u>
5. Week 4 - Data Population	<u>11-12</u>
6. Week 5 - Mid-Project Evaluation	<u>13</u>
7. Week 6 - Indexing and Relational Queries	<u>14-19</u>
8. Week 7 - Embedded SQL Queries	<u>20-22</u>
9. Week 8 - ER Diagram	<u>23-24</u>
10. Week 9,10 - Continuation and Bonus Work	<u>25-32</u>
11. Week 11,12 - Submission and Self Evaluation	<u>33-38</u>

Project Information

Project Title: **Bharat कक्षा**

Github Repository URL: https://github.com/anuneetanand/DBMS_Project

Project Abstract



What can you relate to after seeing this picture, “Government school” is probably the first thing coming to your mind. Do you know that there are about 1 lakh government schools in India and about 1225 Kendriya Vidyalaya schools? These schools follow the same curriculum and guidelines but still lack efficient interconnection. The current methods

for data collection and dissemination from and to these schools are highly inefficient and do not operate under a centralized umbrella, as a result, the new policies and budgets are formulated on the basis of these incomplete statistics which further create a domino effect in hampering the progress of government schools.

Lack of technological tools also paves a way for inefficient school management. We know that only a select number of government schools actually create good outputs despite so much effort by the government.



What if strategists could identify a pattern as to why some schools are creating better results than others?
What if school management could do away with all forms of physical book keeping and allocate more resources towards important services?

Project Proposal

Taking inspiration from the Digital India campaign launched by our Hon'ble Prime Minister Shri Narendra Modi, we aim to design a database management system and a portal to knit together various government schools and maintain their records digitally. It helps in establishing a better relationship between students and higher authorities. The students can use the portal to access books, assignments and exam details. The teachers can use it to analyse student performance and identify problems faced by students. This portal also allows the teachers to directly give an online report to the student instead of conducting a meeting. It also provides parents, ways to monitor their children regularly and not just in parent-teacher meetings. So basically, the government's work becomes more accessible and efficient as it leads to establishing an almost direct relationship with its students.

The data pertaining to exams, grades, attendance, performance, official notices can be accessed by the stakeholders easily and updates can be incorporated in the system faster.

The education ministry can monitor the statistics of all the schools and assign them a ranking. They can also infer a lot of details about student health and performance through interconnected records. All these details are required by the government to launch schemes for the welfare of the students and teachers. The system also provides a feedback system to strengthen the operations of schools.

Code Design Pattern:

Model view controller (MVC)

The idea behind MVC is that each section of your code has a purpose, and those purposes are different. Some of your code holds the data of your app, some of your code makes your app look nice, and some of your code controls how your app functions.

Week 1 - Stakeholders

We have identified the following Stakeholders for our domain application.

The are listed below :-

Student

A student is a significant entity for our project because all we aim is to make a change in the lives of those peers who lack due to scarcity of knowledge and technology. So in our portal, a student is exposed to various features which are essential in his/her daily school life and if combined with technology could lighten their workload and can increase their academic performance.

Guardian

As much it is crucial for the students to have the liberty to design their study patterns, it is equally important for parents to keep track of their children's school life as well as their academic performance. It also helps the parents to establish an open relationship with the school instead of attending semester parent-teacher meetings; here, a parent can analyse his child's performance daily and effectively. A parent can easily enquire about various things.

Teacher

Since a child spends almost half of his day with his/her teachers, therefore, it is necessary that a student can easily communicate with teachers. As a part of the teachers' responsibility, it is essential to analyse the work and progress report of each student in the class. Since it is tedious to keep track of each student, therefore our portal provides many features to teachers that help in understanding students better. A teacher can analyse various details about students.

❑ School Administration

Teachers and students do not complete the requirements of a school as the administration is imperative for managing the requirements of a school. As it is said, better the administration, better the school, so here in our portal, we provide various features to the administration to establish a balanced relationship of control with its students and teachers, thus promoting better education. An administration has various authorities.

❑ Education Ministry

Since all the government schools are interlinked, that is all assignments/exams, or important announcements take place at the same time and day. Therefore, it is important that there is an authority which can interlink all the schools and it does not happen that some schools only get the attention of the government. Every school and its students & teachers should be able represent themselves on a much broader platform. If the ministry is providing some scholarships or any relevant notices, then all students should have an equal chance of availing them. Building on many such parameters, in our portal, the Education Ministry plays a vital role and can carry out various tasks.

Week 2 - Queries and Data Entity

Student

- Check the due date of Exams/Assignments, which are held at all government schools at the same time and day for a particular class.

Data Entities: Student, Exams, Assignments

- Check their attendance to help them keep track of their regularity towards the school.

Data Entities: Student, Attendance

- Check score in all 5 subjects in the last exam conducted.

Data Entities: Student, Grades

- Update feedback of teachers so that better communication can be developed between students and teachers.

Data Entities: Student, Teachers

- Enquire about books of a particular subject.

Data Entities: Student, SubjectInfo

Guardian

- Know the performance of the child, either overall or subject-wise.

Data Entities: Parent, Student, Grades

- Enquire about the fee details.

Data Entities: Parent, Student, Fee_Details

- Know about the ranking of a particular Government School either the school the child is studying in or about other schools where you are considering for a transfer as school transfers are easy in government schools.

Data Entities: Parent, Student, School

- Enquire whether the status of scholarship provided by the government is approved?

Data Entities: Parent, Student



- Providing the feature of updating personal details like address, account number.
Data Entities: Parent
- Check their ward's attendance to help them keep track of their regularity towards the school.
Data Entities: Student, Attendance

Teacher

- To know the grades of a particular student.
Data Entities: Teacher, Student, Grades
- Enquire about the Highest grades or Average grades for a subject.
Data Entities: Teacher, Student, Exam
- Self-evaluating their feedback so that it can help to understand the students' expectations from their teachers.
Data Entities: Teacher
- Uploading regular assignments for practice, since all government schools work on the same pattern; therefore, students can discuss assignments even from different schools.
Data Entities: Teacher, Assignments
- Knowing about the number of submissions of a particular assignment so that the difficulty of assignment can be evaluated.
Data Entities: Teacher, Assignments
- List names of Students (having the same school and class as Teacher) who performed poorly (<30) in the exams.
Data Entities: Teacher, Student, Grades

Administration

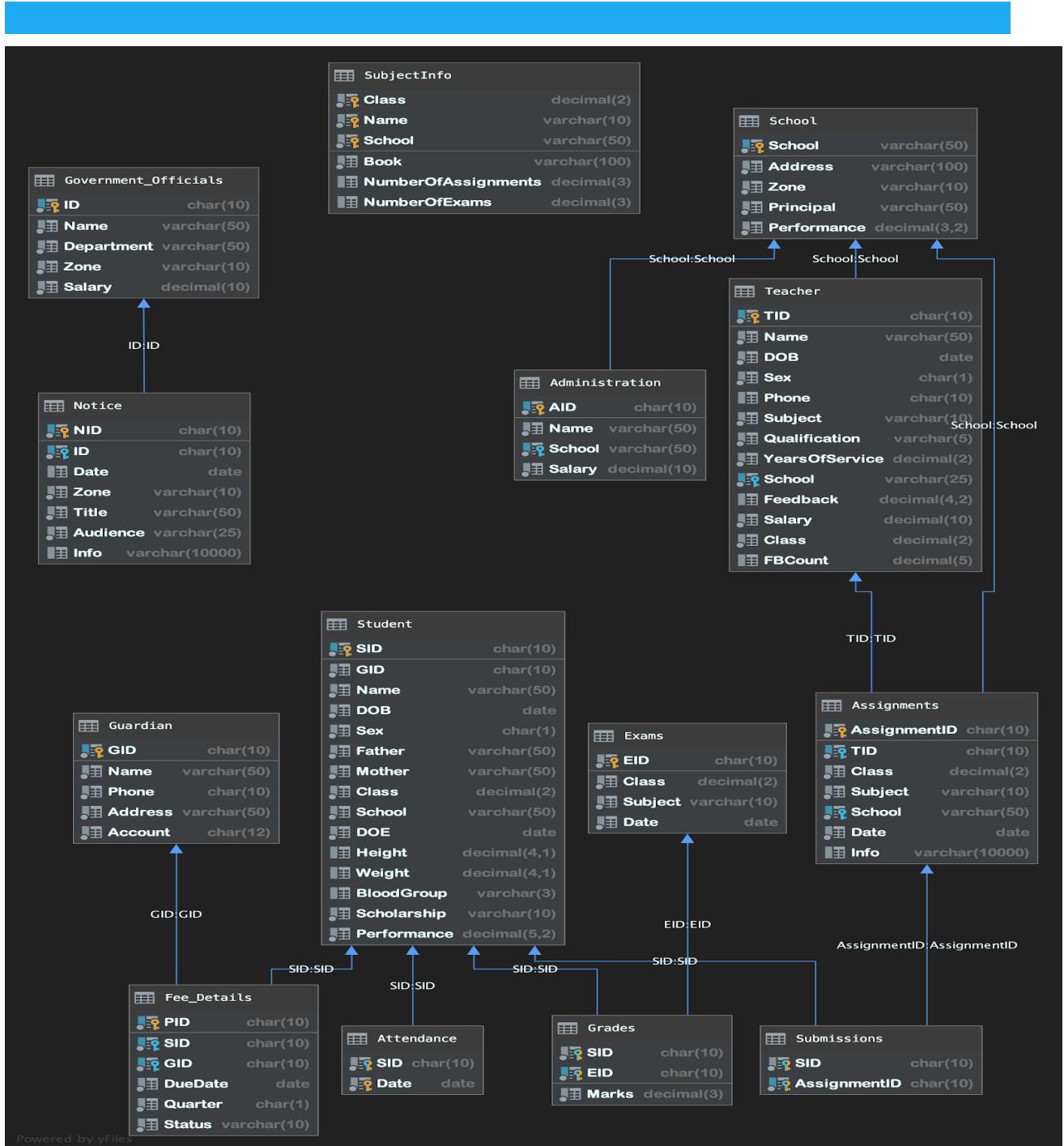
- Enrolling new students (along with their guardians) and assigning them their classrooms and usernames for the portal.
Data Entities: Administration, Student
- Enrolling new teachers for better learning.
Data Entities: Administration, Teachers
- Knowing about their school's performances and rank in comparison to other schools and thus working towards continuous self -evaluation and hard-work to achieve a better rank.
Data Entities: Administration, School

- List the enrollment numbers of Students who have not paid fees.
Data Entities: Administration, Student, Fee_Details
- Knowing about the fee status and updating the fee status(Paid, Due) of a student.
Data Entities: Administration, Student, Fee_Details

Education Ministry

- Issue important announcements or Notices for Students/ Teachers/ Administration.
Data Entities: Government, Notice, School
- Assigning the ranking to its schools, there are almost 326 government schools in Delhi, all of them following the same curriculum and same activities. Therefore each school must assess itself with others.
Data Entities: Government, Administration, School
- Knowing about the students who don't have an appropriate Body Mass Index, it is essential to care about the health of its students' as good health is a critical criterion for measuring happiness and excelling in studies.
Data Entities: Government, Student, School
- Approving Scholarships for students by some cutoff criteria.
Data Entities: Government, Administration, Student
- List Schools with lowest average marks.
Data Entities: Government, School , Grades
- List Schools with lowest average attendance.
Data Entities: Government, School , Attendance
- Check average teacher ratings of a particular school.
Data Entities: Government, School, Teacher

Week 3 - Database Schema



Note: The image have also been uploaded separately in the project folder, but zooming the page also makes the data readable

Week 4 - Data Population

Since our database has different kinds of attributes and varying constraints, therefore we could not generate the data from any online portal. Thus as a resolve, we created about 30 data entries manually for each table and then used python scripts to extend the data up to 1000 entries. We have tried to keep the data as meaningful as possible, but we would still prefer to use and proclaim it as dummy data and future extensions to it would be highly effective and valuable.

Here are a few screenshots that have been taken while compilation of the database which will help in understanding our workflow and requirements :

```

DataPopulation.py •
1  # DBMS Project
2  # Data Population
3
4  from random import randint as R
5  import random
6  import csv
7
8  SID = ["SID"+str(i) for i in range(1000)]
9  GID = ["GID"+str(i) for i in range(1000)]
10 PID = ["PAY"+str(i) for i in range(1000)]
11 TID = ["TID"+str(i) for i in range(1000)]
12 ACC = ["ACTNO"+str(i) for i in range(1000)]
13
14 Phone = [str(R(7000000001,999999999)) for i in range(5000)]
15
16
17 Male = ['Viresh Patel', 'Sampat Patel', 'Prabhakar Singh', 'Mohamad Raj', 'Daman Khan', 'Sevak Raj', 'Janardan
18 Female = ['Arpana Raj', 'Jayalalita Patel', 'Lavali Jain', 'Tripurasundari Singh', 'Manana Raj', 'Gurjari Raj',
19
20 Blood = ['A+','A-','B+','B-','AB+','AB-','O+','O-']
21 Subjects = ['English','Maths','Science','Hindi','GK']
22 Schools = ['SKV Rohini','SKV Karala','SKV Begam Pur','SKV Kanhjawala','SKV Avantika']
23 Student = []
24 Guardian = []
25
26 with open("Student.csv", "w") as csv_file: ...
27 with open("Teacher.csv", "w") as csv_file: ...
28 with open("Guardian.csv", "w") as csv_file: ...
29 with open("Administration.csv", "w") as csv_file: ...
30 with open("GovtOfficial.csv", "w") as csv_file: ...
31 with open("School.csv", "w") as csv_file: ...
32 with open("Notice.csv", "w") as csv_file: ...
33 with open("Exams.csv", "w") as csv_file: ...
34 with open("Assignments.csv", "w") as csv_file: ...
35 with open("Grades.csv", "w") as csv_file: ...
36 with open("Submissions.csv", "w") as csv_file: ...
37 with open("SubjectInfo.csv", "w") as csv_file: ...
38 with open ("Attendance.csv", "w") as csv_file: ...
39 with open ("FeeDetails.csv", "w") as csv_file: ...
40
41 print("Data Generation Successful :)")


:: Line 203, Column 38

```



```
DataPopulation.py
 26 with open("Student.csv", "w") as csv_file:
 27     W = csv.writer(csv_file, delimiter=',')
 28     for i in range(1,601):
 29         Data.append(SID[i])
 30         Data.append(GID[i])
 31         if i%3==0:
 32             Data.append(Male[i])
 33             Student.append(Male[i])
 34             Data.append(str(2015-i//50)+"-0"+str(R(1,9))+"-"+str(R(11,30)))
 35             Data.append("M")
 36             Surname = Male[i].split()[1]
 37         else:
 38             Data.append(Female[i])
 39             Student.append(Female[i])
 40             Data.append(str(2014-i//50)+"-0"+str(R(1,9))+"-"+str(R(11,30)))
 41             Data.append("F")
 42             Surname = Female[i].split()[1]
 43             F = Male[R(1,500)].split()[0]+" "+Surname
 44             M = Female[R(1,500)].split()[0]+" "+Surname
 45             Data.append(F)
 46             Guardian.append(F)
 47             Data.append(M)
 48             Data.append(str(min((i//50+1),12)))
 49             Data.append(Schools[(i%50)//10])
 50             Data.append(str(2019-i//50)+"-04-01")
 51             Data.append(str(150+(i%3)//50+R(1,9)))
 52             Data.append(R(40,80))
 53             Data.append(random.choice(Blood))
 54             if i%20==0:
 55                 Data.append("Yes")
 56             else:
 57                 Data.append("No")
 58             Data.append(R(40,100))
 59             W.writerow(Data)
 60
 61 with open("Teacher.csv", "w") as csv_file:
 62     W = csv.writer(csv_file, delimiter=',')
 63     for i in range(1,301):
 64         Data = []
 65         C = 1*((i-1)%60)//5
 66         Q = ['E','M','H'][C]
 67         Data.append(TID[i])
 68         Data.append(random.choice(Female))
 69         Data.append(str(1980+R(1,10))+"-0"+str(R(1,9))+"-"+str(R(11,20)))
 70         Data.append("F")
 71         Data.append(Phone[i])
 72         Data.append(Subjects[i%5])
 73         Data.append(Q[(i-1)//4])
 74         Data.append(R(1,10))
 75         Data.append(Schools[(i-1)//60])
 76         Data.append(R(1,5))
 77         Data.append(C*5000)
 78         Data.append(C)
 79         W.writerow(Data)
 80
 81 Line 203, Column 38
```

Tab Size: 4 Python

The screenshot shows the MySQL Workbench interface with the 'Student' database selected. The 'Tables' tab displays a table named 'Student' with the following columns and data:

STUDENT_ID	GENDER	NAME	DOB	SEX	FATHER	MOTHER	CLASS	SCHOOL	DOB
SID000000001	GID000000001	Sneepat Patel	2015-03-24	M	Devak Patel	Vandana Patel	1	SKV Rohini	2019-04-0
SID000000002	GID000000002	Prabhakar Singh	2015-05-20	M	Rushil Singh	Shravali Singh	1	SKV Rohini	2019-04-0
SID000000003	GID000000003	Triprasadurandi Singh	2014-03-17	F	Anu Singh	Kusumanal Singh	1	SKV Rohini	2019-04-0
SID000000004	GID000000004	Daman Khan	2015-03-20	M	Churu Khan	Vishaya Khan	1	SKV Rohini	2019-04-0
SID000000005	GID000000005	Sevak Raj	2015-02-24	M	Subhan Raj	Maithili Raj	1	SKV Rohini	2019-04-0
SID000000006	GID000000006	Nivritti Ismail	2014-01-21	F	Sandananda Ismail	Kokila Ismail	1	SKV Rohini	2019-04-0
SID000000007	GID000000007	Karanvir Ismail	2015-04-15	M	Shahrukh Ismail	Pillai Ismail	1	SKV Rohini	2019-04-0
SID000000008	GID000000008	Pavilan Ismail	2015-07-13	M	Jyotirdrab Ismail	Manjubha Tomail	1	SKV Rohini	2019-04-0
SID000000009	GID000000009	Sundari Khan	2014-02-11	F	Sugrev Khan	Putul Khan	1	SKV Rohini	2019-04-0
SID000000010	GID000000010	Daivik Jain	2015-03-11	M	Subodh Jain	Diti Jain	1	SKV Karala	2019-04-0
SID000000011	GID000000011	Asgar Jain	2015-03-11	M	Hasmat Jain	Taraera Jain	1	SKV Karala	2019-04-0
SID000000012	GID000000012	Parnashri Raj	2014-02-23	F	Lalitmohan Raj	Anju Raj	1	SKV Karala	2019-04-0
SID000000013	GID000000013	Ulagan Raj	2015-04-30	M	Budhil Raj	Abhirathi Raj	1	SKV Karala	2019-04-0
SID000000014	GID000000014	Uttara Raj	2015-04-28	M	Sulochana Raj	Indrakshi Raj	1	SKV Karala	2019-04-0
SID000000015	GID000000015	Zaynab Singh	2014-05-15	F	Devans Singh	Chandrakara Singh	1	SKV Karala	2019-04-0
SID000000016	GID000000016	Akshay Singh	2015-03-30	M	Naren Singh	Radhe Singh	1	SKV Karala	2019-04-0
SID000000017	GID000000017	Jugnu Patel	2015-04-11	M	Nabarun Patel	Kiran Patel	1	SKV Karala	2019-04-0
SID000000018	GID000000018	Yosana Ismail	2014-04-26	F	Basudha Ismail	Gangotri Ismail	1	SKV Karala	2019-04-0
SID000000019	GID000000019	Prabodh Kumar	2015-09-16	M	Jugnu Kumar	Seren Kumar	1	SKV Karala	2019-04-0
SID000000020	GID000000020	Narvox Ismail	2015-01-14	M	Pratul Ismail	Sukeshi Ismail	1	SKV Begam Pur	2019-04-0
SID000000021	GID000000021	Neeharika Jain	2014-06-19	F	Jitendra Jain	Vishaya Jain	1	SKV Begam Pur	2019-04-0
SID000000022	GID000000022	Fadi Patel	2015-09-30	M	Saeed Patel	Trilyama Patel	1	SKV Begam Pur	2019-04-0
SID000000023	GID000000023	Dilys Patel	2015-05-23	M	Sharmila Patel	Varsha Patel	1	SKV Begam Pur	2019-04-0
SID000000024	GID000000024	Shivani Kumar	2014-01-14	F	Anu Kumar	Chareesha Kumar	1	SKV Begam Pur	2019-04-0
SID000000025	GID000000025	Jaywant Patel	2015-01-11	M	Patakin Patel	Ranjini Patel	1	SKV Begam Pur	2019-04-0
SID000000026	GID000000026	Premaa Singh	2015-01-20	M	Gulab Singh	Trilochana Singh	1	SKV Begam Pur	2019-04-0
SID000000027	GID000000027	Mathilini Khan	2014-09-23	F	Kairisha Khan	Chintana Khan	1	SKV Begam Pur	2019-04-0
SID000000028	GID000000028	Mirri Singh	2015-06-10	M	Brahmabrat Singh	Kadambini Singh	1	SKV Begam Pur	2019-04-0
SID000000029	GID000000029	Talleen Raj	2015-05-27	M	Vishwajit Raj	Chameli Raj	1	SKV Begam Pur	2019-04-0
SID000000030	GID000000030	Trikaya Patel	2014-07-30	F	Ulagan Patel	Triprasadurandi Patel	1	SKV Kanjhawala	2019-04-0
SID000000031	GID000000031	Chanchal Patel	2015-08-27	M	Sachchidananda Patel	Hariananti Patel	1	SKV Kanjhawala	2019-04-0
SID000000032	GID000000032	Susandu Ismail	2015-09-16	M	Rajen Ismail	Pradeep Ismail	1	SKV Kanjhawala	2019-04-0
SID000000033	GID000000033	Omara Raj	2014-05-27	F	Haibb Raj	Rukhsana Raj	1	SKV Kanjhawala	2019-04-0
SID000000034	GID000000034	Selma Ismail	2015-01-20	M	Kandarpore Ismail	Vishwanya Ismail	1	SKV Kanjhawala	2019-04-0
SID000000035	GID000000035	Hishmishi Jash	2015-03-20	M	Salem Jash	Yashini Jash	1	SKV Kanjhawala	2019-04-0
SID000000036	GID000000036	Aparigita Patel	2014-05-29	F	Vineet Patel	Chitrakarsha Patel	1	SKV Kanjhawala	2019-04-0
SID000000037	GID000000037	Vishak Kumar	2015-02-11	M	Shyamsunder Kumar	Kerunya Kumar	1	SKV Kanjhawala	2019-04-0
SID000000038	GID000000038	Balagovind Ismail	2015-05-26	M	Iasivalan Ismail	Kadambini Ismail	1	SKV Kanjhawala	2019-04-0
SID000000039	GID000000039	Sedhvi Patel	2014-07-18	F	Harsh Patel	Sharvani Patel	1	SKV Kanjhawala	2019-04-0
SID000000040	GID000000040	Tralkova Patel	2015-03-14	M	Mohak Ismail	Udita Ismail	1	SKV Avantika	2019-04-0
SID000000041	GID000000041	Tralkova Patel	2015-06-11	M	Mohamed Patel	Vinanti Patel	1	SKV Avantika	2019-04-0
SID000000042	GID000000042	Shatrupsa Raj	2014-01-17	F	Pracheeta Raj	Akshithi Raj	1	SKV Avantika	2019-04-0
SID000000043	GID000000043	Sabrina Patel	2015-01-17	M	Komy Patel	Indra Patel	1	SKV Avantika	2019-04-0
SID000000044	GID000000044	Anilash Raj	2015-02-27	M	Chandrakanta Raj	Devaviree Raj	1	SKV Avantika	2019-04-0
SID000000045	GID000000045	Agrata Raj	2014-08-24	F	Hireesh Raj	Mehati Raj	1	SKV Avantika	2019-04-0
SID000000046	GID000000046	Chandresh Jain	2015-09-28	M	Gunjan Jain	Arushi Jain	1	SKV Avantika	2019-04-0

500 rows retrieved starting from 1 in 431 ms (execution: 12 ms, fetching: 419 ms)

Event Log: 3:1 Material Darker

Note: The images have also been uploaded separately in the project folder, but zooming the page also makes the data readable.

Week 5 - Mid - Project Evaluation

We completed our Mid-Project evaluation in the given week, and all group members were active in the assessment. This week we completed all the pending work required for the project evaluation that is from documentation to database creation. Further, we planned how to proceed with our work in the next couple of weeks for the project completed successfully in time.

For the rest of the project, we planned that:

- Srijan would work on front-end and organisation.
- Anuneet would code the back-end and link the SQL efficiently to the main program and design the necessary algorithms.
- Himanshi would work on SQL and Relational Algebraic queries and documentation of the Work in Progress document.
- Vikas & MD. Talib will populate the data and look for flaws.

Week 6 - Indexing and Relational Queries

Database Normalization

There are three types of anomalies that occur when the database is not normalized. These are – Insertion, update and deletion anomaly. To avoid these anomalies we tried to normalize the data upto 3NF, firstly there is made sure that all attributes can hold only atomic values (even for address) that is, we do not store multiple values for any attributes in the schema. We know that an attribute that is not part of any candidate key is known as non-prime attribute. So for converting the data we made sure that No non-prime attribute is dependent on the proper subset of any candidate key of table and that is how we achieved the data to be in 2NF. Now further achieving our goals we focused upon reaching the 3NF normalisation, which as claimed by Sir in class is “Most practiced and suitable for companies”, because if we had tried to reduce it to further normalizations it would have resulted in more number of tables and more database storage on backend. So for 3NF, Transitive functional dependency of non-prime attribute on any super key should be removed, as we know that an attribute that is a part of one of the candidate keys is known as a prime attribute. So, therefore our DataBase Design achieves the **3NF** normalisation which is most suited and well optimised according to our designed schema.

Changes done -

- Remove the School Departments Table, and directly connect teachers to the Administration.
- Remove TID(Teacher Primary Key) from Administration Table.
- Create a new Table Subject Indo to be able to Class details from Student, since every class will have the same details.

- Added books attribute as well to SubjectInfo, so that if any student tries to search for books it can be fetched in retrieving with the class name and not anything else.
- Removed percentage from attendance, to be able to calculate it as a function whenever required and not stored it unnecessarily.
- Stored a student's personal information in Guardian Table only and did not repeat the information from both sets.

Indexing

Indexing plays an important role in optimizing the performance of any database. It helps to quickly locate and access the data. Creating an index on a field in a table of the database creates another data structure which holds the field value and a pointer to the record it references. Sorting our data based on the index allows database engines to perform Binary Search to speed up query processing.

Our application is based on the database of students, teachers and many other entities involved in the education sector of society. We acknowledge the fact that there are a large number of government schools in India. In order for the efficient performance of our application, we created the following Indexes for our Relations.

[Please refer to the Relation Schema for more information about attributes.]

Table Name	Cluster of Attributes Indexed	Description
Assignments	TID, Class, School	To search for assignments given to class wise and school wise.
Administration	AID	To fetch all administration of different schools uniquely.
Fee_Details	PID, GID, SID	For parents to know the fee status of their ward in minimal time.
Attendance	SID	To be able to uniquely point to the attendance attribute for each student.
Exams	EID	To store all exam ids with their classes and subjects uniquely.
Grades	SID, EID	To retrieve grades of students directly by going to the exam id.

Notice	NID, ID	To be able to reach the notice head that is the government uniquely.
Government_Officials	ID	To be able to store all government heads particularly.
Guardian	GID	To fetch all personal records of guardians as well as students.
School	School	To be able to determine from School name uniquely.
SubjectInfo	Name	To fetch all books and class details for a subject.
Submissions	AssignmentID, SID	To be able to calculate the number of assignments done for each student efficiently.
Student	SID	To fetch all records of students precisely.
Teacher	TID, School	To be able to directly locate Teacher's School because we have records for multiple government schools.

Relational Algebraic Queries

1) Parent enquiry for fee details .

$$\rho(Fee, Fee_Details)$$

$$\pi_{Fee.DueDate}(\sigma_{Fee.GID=InputID}(Fee))$$

2) Parent enquiring about scholarship status.

$$\rho(S, Students)$$

$$\rho(G, Guardian)$$

$$\pi_{S.Scholarship}(\sigma_{S.SID=G.SID}(S \times \pi_{G.SID}(\sigma_{G.PID=Input.PID}(G))))$$

3) Parents know the highest Ranking School.

$\rho (SC, School)$

$\rho (S, Student)$

$(\pi_{sc.performance} (SC) - \pi_{sc.performance} (SC \bowtie_{Performance < Performance2}$

$(\rho_{Performance2/Performance} SC))) \times$

$\pi_{S.School} (\sigma_{S.SID=G.SID} (Student \times \pi_{G.SID} (\sigma_{G.PID=Input.PID} (G$

)

4) Teachers self evaluate their feedback.

$\rho (T, Teacher)$

$\pi_{T.Feedback} (\sigma_{T.TID=GivenID} (T))$

5) Teaching knowing about the number of submissions.

$\rho (A, Assignment)$

$\rho (T, Teacher)$

$\pi_{A.NoOfSubmissions} (\sigma_{A.TID=T.TID} \times \pi_{TID} (\sigma_{TID=GivenID} (T)))$

6) Students checking Due Date for Assignments.

$\pi_{Assignment.DueDate} (\sigma_{Assignment.AID=GivenID} (Assignment)))$

7) Administration knowing about the fee status of students.

$\rho (F, fee_details)$

$\rho (S, Students)$

$$\pi_{F.DueDate} (\sigma_{F.SID=S.SID} (F \times \pi_{S.SID} (\sigma_{S.Name=Input.Name} (S))))$$

)

8) Government reading information about notices.

$\rho (N, Notices)$

$\rho (G, Government_officials)$

$$\pi_{N.Title, N.Info} (\sigma_{N.ID=G.ID} (N \times \pi_{G.ID} (\sigma_{G.ID=Input.ID} (G))))$$

9) Government taking list of all Teachers from School in post PGT.

$\rho (T, Teacher)$

$\rho (S, School)$

$$\pi_{T.Post} (\sigma_{T.Post="PGT"} (Teacher \times \pi_{S.School} (\sigma_{S.School=Input.School})))$$

10) Student knowing about attendance.

$$\pi_{Attendance.Date} (\sigma_{Attendance.SID=InputID} (Attendance)))$$

11) Students know the books required .

$\rho (SB, SubjectInfo)$

$\rho (S, Student)$

$$\pi_{SB.Book} (\sigma_{SB.Class=S.Class} (SB \times \pi_{S.Class} (\sigma_{S.SID=GivenID} (S))))$$

)

12) Administration list of Students with unpaid fees.

$$\begin{aligned} \pi_{S.Name} (\sigma_{S.SID=P.SID} (Student \\ \times (\pi_{Fee_Details.SID} (\sigma_{Fee_Details.Status="UNPAID"} (Fee_Details))) \end{aligned}$$

13) Student updating teacher feedback.

$$\begin{aligned} \rho(T, Teacher) \\ T \leftarrow \pi_{T.Feedback=InputGiven} (\sigma_{T.TID=InputTeacher.TID} (T)) \end{aligned}$$

14) Government approving Scholarship by some Cutoff.

$$\begin{aligned} \rho(S, Student) \\ S \leftarrow \pi_{S.SID, S.Scholarship="YES"} (\sigma_{S.Performance > CutOff} (S)) \end{aligned}$$

15) Admin enrolling a new Teacher

$$\begin{aligned} Teacher \leftarrow Teacher \cup \{All\ Attributes\ of\ Teacher\} \\ Assignment \leftarrow Assignment \cup \{Adding\ new\ TID\} \end{aligned}$$

16) Admin enrolling a new Student

$$Student \leftarrow Student \cup \{All\ attributes\ of\ Student\}$$

```
Grades ← Grades ∪ {Adding SID}  
Fee_Details ← Fee_Details ∪ {Adding SID}  
Attendance ← Attendance ∪ {Adding SID}
```

Week 7 - Embedded SQL Queries

The project has been designed in Java using JavaFX for the frontend. In order to work with the database, we used mysql-connector for java and JDBC drivers. The queries have a general form involving the following entities :

- **Prepared Statement [or Statement]** : SQL query framed as a string in java, say sql
- **executeQuery(sql) [or executeUpdate(sql)]** : To send the query/update to the database.
- **ResultSet** : It holds the values of the relation returned by the database. Individual columns of a row can be accessed by specifying their index in getString() method.

Some of the embedded queries have been presented below. [Please refer code for more details.]

To Retrieve Student Info

```
"SELECT * FROM STUDENT WHERE SID = " + ID + ";"
```

To Retrieve Assignment Info

```
SELECT AssignmentID,Subject,Date,Info FROM Assignments WHERE AssignmentID IN  
(SELECT AssignmentID from Assignments where Date>CURDATE() AND Class="" + Class  
+ " and School="""+School+""");;
```

To Submit Feedback

```
"UPDATE Teacher SET FBCount = FBCount + 1 WHERE ( Class = "" + Class + "" AND School  
= "" + School + "");;  
"UPDATE Teacher SET Feedback = ((Feedback * 10)+" + EF.getText() +")/FBCount WHERE (  
Class = "" + Class + "" AND School = "" + School + "" AND Subject = 'English' );;";
```

To Analyse Marks

```
"select AVG(Marks), MAX(Marks),MIN(Marks) from Grades JOIN Exams E on Grades.EID =  
E.EID WHERE E.Class = "+Class+" AND E.Subject = "+Subject+" AND Grades.SID in (  
SELECT SID FROM Student where School = "+School+");";
```

To Add New Assignment

```
"INSERT INTO Assignments values(?,?,?,?,?,?)";
```

To Update Details

```
"UPDATE Student SET Height = "+Height.getText()+" , Weight = "" + Weight.getText() + "  
WHERE GID = "+GID+";";;
```



To Retrieve Marks

"SELECT Marks,Subject from Grades JOIN Exams E on Grades.EID = E.EID WHERE Grades.SID IN (SELECT SID FROM Student where GID = "+GID+");";

To Assign Ranks to Schools

" SELECT School, RANK() over (order by Performance) AS R from School;";

To List Students Who Have Their Fee Due

"SELECT S.SID from Fee_Details JOIN Student S on Fee_Details.SID = S.SID WHERE School = " + School + " AND Status = 'Due';";

To Update Performance Score Of School

"UPDATE School SET Performance = " + Score.getText() + " WHERE School = "+SchoolName.getText()+";";

To Calculate Average BMI Of A Class Across All Schools

"SELECT School,ROUND(AVG(Weight*10000/(Height*Height)),2) AS x FROM Student where Class = "+AnalyseClass.getText()+" group by School;";



To Calculate Average Attendance Of A Class Across All Schools

```
"SELECT School,AVG(A)*5 as x from (SELECT COUNT(Student.SID) AS A,Student.SID,Class,School from Student JOIN Attendance A on Student.SID = A.SID group by Student.SID) as SA where Class = "+AnalyseClass.getText()+" group by School;";
```

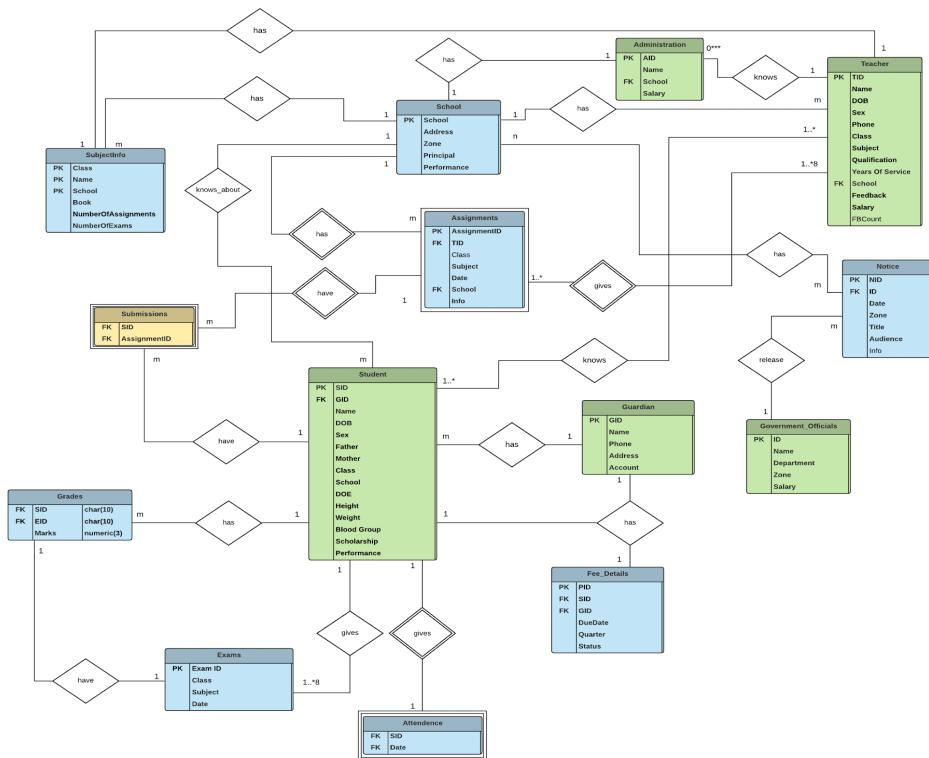
More in Code...

Week 8 - ER Diagram

[https://www.lucidchart.com/documents/edit/aaa1fb25-9a74-4722-9f29-c4fb1319ab98/0_0
?shared=true](https://www.lucidchart.com/documents/edit/aaa1fb25-9a74-4722-9f29-c4fb1319ab98/0_0?shared=true)

ER DIAGRAM FOR SCHOOL DATABASE

TEAM-76



Note: If you found that image is not readable then by zooming this page also makes it readable and we have also provided the link above the diagram.

ER Diagram Description

- The above ER Diagram mainly consists of 14 entities. That are School, Administration, Teacher, Student, Subject_Info, Assignments, Submissions, Exams, Grades, Guardian, Notice, Government officials, Fee Details and Attendance

- Strong entities are Student, Teacher, School, Guardian, Exams, Grades, Subjectinfo, Government officials, Notice and Fee details.
- Weak entities are like Submissions, Attendance and Assignments.
- There can be some Composite attributes like Address(Street, City, State), Name(first, Second and Last name) but not had made them composite.
- After that Simple attributes are like Class, Height, Blood group, etc(Because they can not be divided further).
- Now Single-Valued attributes are like D.O.B, Sex, Father ,Mother, etc(Because they can take only single value)
- After that Multivalued attributes are like Phone no. (one guardian may have more than one phone no.) The multivalued attributes can have more than one value or skills as we mentioned earlier. Again we had not made any attribute multivalued.
- The solid lines that connect attributes to show the relationships of entities in ER diagram.
- In this diagram we used actions which are represented by diamond shapes, which shows how two entities share or link information to each other in the database.
- Cardinality has also an important role in ER diagram and In entity relationship diagram,we have written cardinality above the lines.
- It specifies how many instances or generations of an entity can relate to one instance or generation of another entity.
- Basically it specifies the occurrence of a relationship of an entity that is the maximum number of relationships of an entity.
- So finally, ER diagram has mainly three main components : Entity, Attributes and Relationship among them.
- Entities can be strong or weak depending on the condition of parameters. And Attributes are mainly composite, key, multivalued, derived. Relation between them matters most which can be of types one-one represented by 1-1, one to many represented by 1-n or 1-m, many to many represented by m-n, many to one represented by m-1 or n-1.
- We have used different colours for important parts in the ER diagram.

Week 9,10 - Continuation with Bonus Implementation



In the given week being quite close to the deadline, we started to pace our work by having frequent group meetings and therefore decided to achieve the set targets along with bonus implementation. We began to work upon an appealing front end that enhances our sharp and well-embedded backend. While integrating the backend and frontend was quite tiresome, we tried to achieve a suitable portal with attractive features and a robust suite to attract our stakeholders by providing them with an excellent interface along with exciting and necessary features.

Design Documentation (Front End)

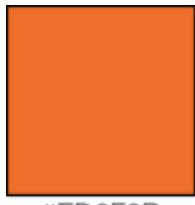
Software Used: Adobe XD (Experience Design)

The front has been crafted using [Google's Material Design Guidelines](#) for maximum usability and enhanced user-experience.

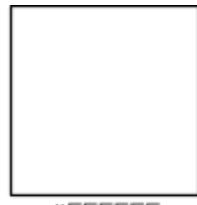
Material Design is a visual language that synthesizes the classic principles of good design with the innovation of technology and science. It is inspired by the physical world and its textures, including how they reflect light and cast shadows. Material surfaces reimagine the mediums of paper and ink.

Moreover, the site navigation and the information architecture for the application has been developed while keeping a human-centric approach to ensure ease of access and give the user an intuitive feel.

Colors



#ED6F2B



#FFFFFF



#525A65

Keeping our target audience in mind, we have chosen hue of Orange as our primary color because it is representative enthusiasm, fascination, happiness, creativity, determination, attraction, success and encouragement.

Fonts

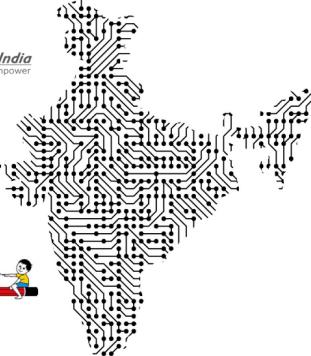
Roboto

Roboto has a dual nature. It has a mechanical skeleton and the forms are largely geometric. At the same time, the font features friendly and open curves. While some grotesks distort their letterforms to force a rigid rhythm, Roboto doesn't compromise, allowing letters to be settled into their natural width. This makes for a more natural reading rhythm more commonly found in humanist and serif types.

Screens:



Taking inspiration from the Digital India campaign launched by our Hon'ble Prime Minister Shri Narendra Modi, we aim to design a database management system and a portal to knit together various government schools and maintain their records digitally. It helps in establishing a better relationship between students and higher authorities. The students can use the portal to access books, assignments and exam details. The teachers can use it to analyse student performance and identify problems faced by students. This portal also allows the teachers to directly give an online report to the student instead of conducting a meeting. It also provides parents, ways to monitor their children regularly and not just in parent-teacher meetings. So basically, the government's work becomes more accessible and efficient as it leads to establishing an almost direct relationship with its students.



MEET THE TEAM

Facilities



Connecting
Connecting millions of parents, students and teachers across the country.



Cloud Storage
Providing cloud data storage services to thousands of schools across India.



PerformX™
Analytics to track school performance based on region.



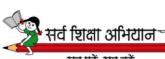
Assignments
Assignment retrieval and submission for students



E-Payment Portal
Pay school fee online without any hassle



हमारे माननीय प्रधान मंत्री श्री नरेंद्र मोदी द्वारा शुरू किए गए डिजिटल इंडिया अभियान से प्रेरणा लेते हुए, हमारा उद्देश्य विभिन्न सरकारी स्कूलों को एक साथ बुनाना और उनके रिकॉर्ड को डिजिटल रूप से बनाए रखने के लिए एक डेटाबेस प्रबंधन प्रणाली और एक पोर्टल तैयार करना है। यह छात्रों और उच्च अधिकारियों के बीच बहतर संबंध स्थापित करने में मदद करता है। छात्र पुस्तकों, असाइनमेंट और परीक्षा विवरण तक पहुंचने के लिए पोर्टल का उपयोग कर सकते हैं। शिक्षक छात्र के प्रदर्शन का विश्लेषण करने और छात्रों के समान आने वाली समस्याओं की पहचान करने के लिए इसका उपयोग कर सकते हैं। यह पोर्टल शिक्षकों को बैठक आयोजित करने के बजाय सीधे छात्रों को ऑनलाइन रिपोर्ट देने की अनुमति देता है। यह माता-पिता, नियमित रूप से और न केवल माता-पिता-शिक्षक बैठकों में अपने बच्चों की निगरानी के तरीके प्रदान करता है। इसलिए मूल रूप से, सरकार का काम अधिक सुलभ और कुशल हो जाता है क्योंकि यह अपने छात्रों के साथ लगभग सीधा संबंध स्थापित करने की ओर जाता है।



टीम से मिलो

सुविधाएं



कनेक्ट
देश में लाखों माता-पिता, छात्रों और शिक्षकों को जोड़ना।



बादल भंडारण
पूरे भारत के हजारों स्कूलों में क्षेत्रफल स्तर पर संवर्तन करना।



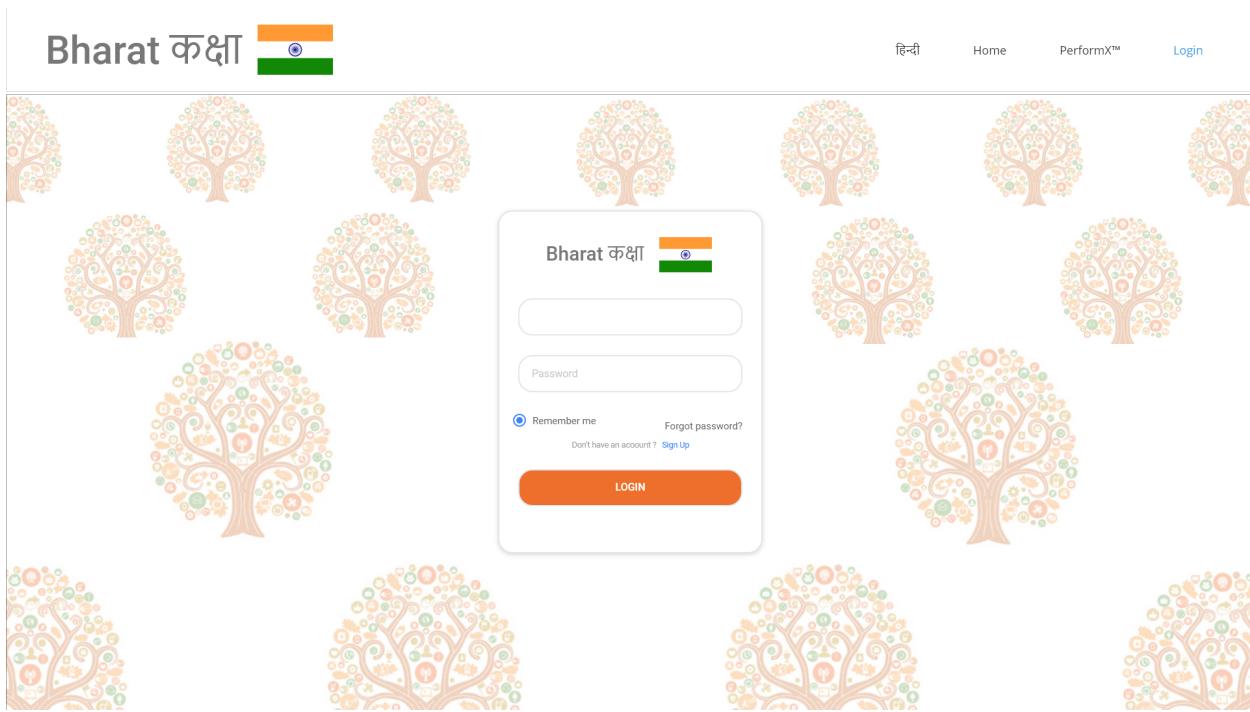
प्रदर्शनX™
छात्र के जीवन पर क्षेत्र के प्रदर्शन को ट्रैक करने के लिए विश्लेषित।



कार्य
छात्रों के लिए अलग-अलग पुनर्वित्तन और प्रस्तुत करना।



ई-पेमेंट पोर्टल
विना किसी परेशानी के अन्तराल में इच्छुक परीक्षा का भुगतान करें।



The student profile screen for Srijan Jain (ST2018319) displays various performance metrics. The left sidebar shows placeholder address and contact information. The main area shows attendance at 90%, two books assigned, and zero notices. The grades section shows English at 90%, Hindi at 90%, Maths at 70%, Science at 96%, and GK at 96%. The assignments due section is empty. The feedback section shows two pending feedback items for English, Hindi, Maths, and Science. A large orange 'SUBMIT' button is located at the bottom right.

Attendance	Books	Notices
90%	2	0

Grades		Assignments Due		Feedback	
English :	90%			English :	2
Hindi :	90%			Hindi :	2
Maths :	70%			Maths :	2
Science :				Science :	2
GK :	96%			GK :	2



Prof. Mohd. Shah

TEACHER : ST2018319

<Address>
<Address>
<Address>

<Guardian Name>

<Emergency Contact Number>

POST ASSIGNMENT

Description

Due date

Attach file

SUBMIT

SUBMISSIONS

PERFORMANCE

Highest : 95%**Average : 95%****Lowest : 95%**

WEAK STUDENTS

RATINGS

0

<GOVT>

GOVERNMENT :

<Address>
<Address>
<Address>

<Emergency Contact Number>

CREATE NEW NOTICE

TEXT

Audience

ISSUE

TEACHER RATINGS

APPROVE SCHOLARSHIP

SID

UPDATE

UPDATE SCHOOL PERFORMANCE

SID

Performance

UPDATE

DEMOGRAPHICS



Bharat कक्षा



हिन्दी

Home

PerformX™

Logout

<GUARDIAN>

GUARDIAN :

<Address>
<Address>
<Address>

<Ward Name>

<Emergency Contact Number>

FEE DETAILS

FEE

Secure Payment



PAY NOW

SCHOOL RANKINGS

WARD PHYSICAL DETAILS

Height (in cm)

Weight (in Kgs)

UPDATE

PERFORMANCE

English :
Hindi :
Maths :
Science :
GK :
Overall :
(calculated by standard mean)

ATTENDANCE

83%

SCHOLARSHIP

Approved!

Bharat कक्षा

हिन्दी

Home

PerformX™

Login

ANALYTICS

SID

LOGIN

SCORE PREDICTOR

BMI

Average attendance

Average Marks

Average teacher feedback

PREDICT

PREDICTED SCORE : **96%**



Access Restricted



Fundamentals of Database Management Systems (CSE-202)

Bharat कक्षा



Team 76

Anuneet Anand | Himanshi Mathur | Srijan Jain | MD. Talib | Vikas Mahto

We consider the following features, which are a part of **Bonus Implementation**; that is, these features made our project much more delightful and up to the mark along with right-thinking achieved towards our aim of Digital India.

The Bonus Features are described as following:

Minor but important things taken care of :-

- All login pages have been created to assure that no stakeholder can access each other's data and data security is ensured.
- Database is normalized to 3NF which ensures our database to be well optimised and productive.
- We have included multiple insert/ update queries by taking care of all the data anomalies arised.
- We ensured that we have formalized only one connection with SQL in our database and accordingly the other further login page arrives, thus making it more optimized and reducing conflicts.

Extensive and Innovative Component :-

- Material Design UI
- Interactive Graph
 - To give a visual analysis for the data
- Hindi Language Support
 - To increase accessibility
- Restricted Access & Privacy to analytics
 - To ensure data security and prevent misuse
- Score Prediction
 - Probabilistic formulas to determine future success of a student
- Smart Login
- Error Handling
 - To prevent crashing of system and protect the system from user faults

(More during demo)

Week 11,12 - Submission and Self Evaluation

Future Extensions

- Working on a voice recognition system and to be able to build a more user friendly atmosphere for our stakeholders.
- By using many APIs provided to access another company's data, piece of code, software, or services, and extend the functionality of our product while saving time and money because APIs do all the heavy lifting in the background, digital experiences remain virtually effortless.
- Additionally, the extension will allow us to assist some of our beneficiaries in organizing their own networking sessions to their experiences from our project.
- Our challenge now is to move forward to the next stage, seeking to strengthen existing impacts and further consolidate their usefulness and effectiveness.
- Towards this end, we plan to submit to you soon a final narrative outlining what we have learned from the project so far.

Work Distribution

Week 1

- Deciding the domain to work upon for the project - All members
- Deciding an innovative Team Name - Anuneet, Talib, Vikas
- Deciding the Stakeholders for the domain - Anuneet, Himanshi, Srijan

Week 2

- Thinking and writing down the 5 queries for the 5 Stakeholders that will constitute the domain questions of our system.
 - ◆ Student - Anuneet
 - ◆ Teacher - Himanshi
 - ◆ Guardian - Srijan
 - ◆ Administration - Talib
 - ◆ Education Ministry - Vikas

Week 3

- Designing the database schema with 5 main tables and rest supporting tables required to build our portal for advanced education.
 - ◆ Student - Anuneet
 - ◆ Teacher - Himanshi
 - ◆ Guardian - Srijan
 - ◆ Administration - Talib
 - ◆ Education Ministry - Vikas
 - ◆ Supporting tables - Anuneet, Himanshi

Week 4

- Data Population - Himanshi
- Data Mining - Srijan, Talib, Vikas



- Organising and Importing Data - Anuneet

Week 5

- Working on completing the previous week's documentation - Himanshi, Srijan, Anuneet
- Presenting the Mid evaluation and further discussion of ideas - All members

Week 6

- Redesigning the database to normalize it to 3NF - All members
- Creating the indexes
 - ◆ Selection of Attributes - Anuneet, Himanshi
 - ◆ Discussion over Attributes - Srijan, Vikas
 - ◆ Implementing these queries - Anuneet
- Writing Relational Algebra Queries - Himanshi, Talib

Week 7

- Thinking and Discussing upon Embedded SQL Queries - All members
- Implementation - Anuneet
- Working on side by side documentation and Relational Queries - Himanshi

Week 8

- Deciding the layout of ER diagram - All members
- Mapping the ER diagram on LucidChart - Vikas, Talib
- Laying down parameters for further backend and frontend work - Anuneet, Himanshi, Srijan

Week 9

- Front End Basic Layout - All members

- Creating and Designing Home Page, Login Page Student, Login Page Guardian - Srijan
- Creating Login Page Administration, Creating Login Page Guardian - Himanshi
- Creating Login Page Government Officials - Talib
- Working on Query optimisation - All members
- Integrating query optimisations - Anuneet

Week 10

- Integrating backend and frontend - Anuneet , Srijan
- Thinking and building upon bonus components - All members
- Completing documentation with all the requirements - Himanshi , Vikas

Week 11 - 12

- All of us worked hard to complete our project Bharat kaksha and to achieve our final goal ; an online platform to help the underprivileged students get an equal opportunity for digitalised education and to be able to achieve success in life.
- Working on self Evaluation - All members

Self Evaluation

Student Name	Score (out of 100)	Remarks
Anuneet	100	I had been really passionate about this project from the very beginning. I put in a considerable amount of effort to design an effective backend on Java platform. My teammates made it possible to develop the overall project in an effective manner. This is our step towards Digital India.
Himanshi	100	I believe that in this project we have given our heart and soul and completed all the requirements and deliverables in addition to implementing bonus features (even thinking of minute optimisations while framing the database), and wholeheartedly and gave as much time as possible to the project. We have tried to build an exciting and astonishing portal for digital education. Also, future extensions proposed to the project will help the project to be one of its kind.
Srijan	100	This project has been something really close to my heart. For the first time during my degree at IIITD, I have worked on something so extensively from front end to backend. I believe that this project can actually be extended on a large scale with minor updates and changes. This project provided me with an opportunity to work with some very talented people in our college. It was a great team effort overall!
Talib	100	In this project, I have been worked hard to complete this interactive project. We all are worked hard to achieve our goal. Team members are very supportive they had also help to made this interactive online platform which can manage most of problems like school department and administration of school are faced lots of problem like what is the progress of students ? How can we overcome the difficulties of most of the possible maximum students ? We have made digital environment for them so that they can easily manage the problem and reduce it to as possible as they can. And students also get the equal opportunity for this digital database. By the way it was

		great experience with our team member. I liked it the most.
Vikas	100	This project is one of the most organised projects I have done in our college.In this project i had mainly dealt with the ER Diagram.This project also provided me a good workspace to enhance my knowledge. Every team member had given its best towards this project.And one thing that goes in favour of this project is that it deals with the real life problem that if faced by lakhs of students, teachers thousands of schools everyday. Atlast it was a very good experience to work with our brilliant minded team mates.