

Описание

Электронный журнал

Проект по созданию электронного журнала с применением базы данных.

Основная задача:

Создание необходимого функционала для информирования учеников (или родителей) об их успеваемости, а также создание необходимых условий для дистанционного редактирования данных учителем. В связи с этим необходимо перенять лучшие качества у бумажного варианта журнала и оптимизировать их для пользователей.

Данные

User:

`id_user`: INT, primary key, not null, unique index, unsigned data type, auto incremental;

`fio`: VARCHAR(64), not null;

`isteacher`: TINYINT;

User_has_Class:

`User_id_user`: INT, primary key, not null, foreign key (referenced table `User` column `id_user`);

`Class_id_class`: INT, primary key, not null, foreign key (referenced table `Class` column `id_class`);

Class:

`id_class`: INT, primary key, not null, unique index, unsigned data type, auto incremental;

`class_name`: VARCHAR(32), not null;

User_has_Subject:

`User_id_user`: INT, primary key, not null, foreign key (referenced table `User` column `id_user`);

`Subject_id_subject`: INT, primary key, not null, foreign key (referenced table `Subject` column `id_subject`);

Subject:

`id_subject`: INT, primary key, not null, unique index, unsigned data type, auto incremental;

`subject_name`: VARCHAR(32), not null;

Marks:

`id_mark`: INT, primary key, not null, unique index, unsigned data type, auto incremental;

`User_id_user`: INT, primary key, not null, foreign key (referenced table `User` column `id_user`);

`mark`: CHAR(1);

`Lesson_id_lesson`: INT, primary key, not null, foreign key (referenced table `Lesson` column `id_lesson`);

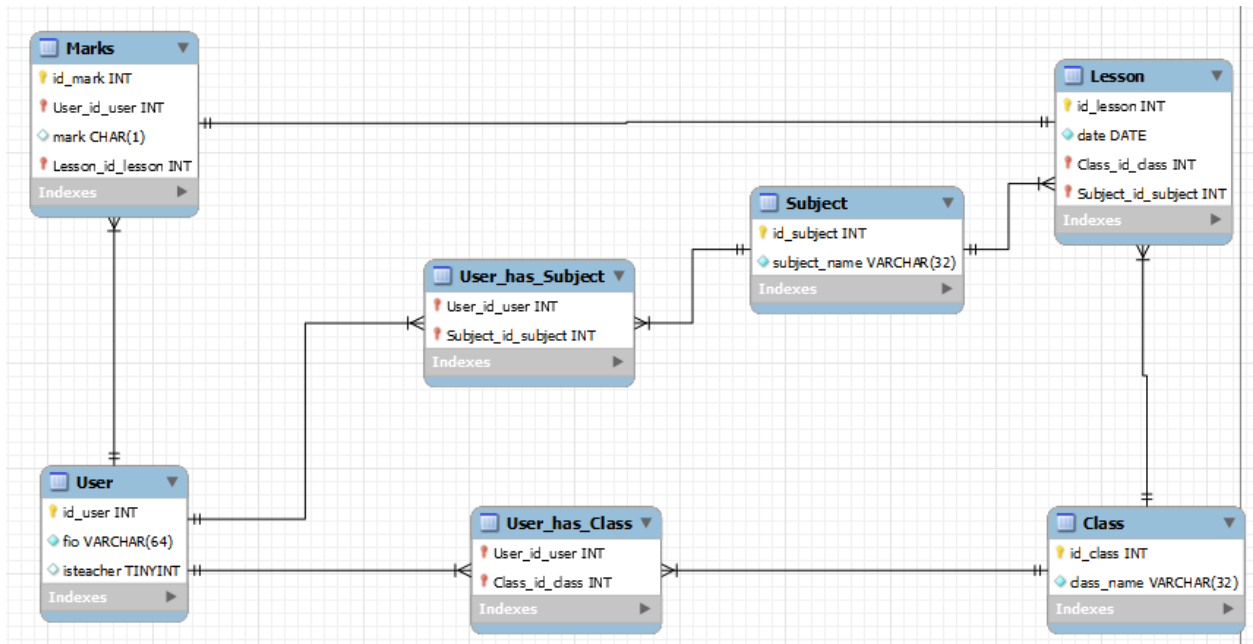
Lesson:

id_lesson: INT, primary key, not null, unique index, unsigned data type, auto incremental;

date: DATE, not null;

Class_id_class: INT, primary key, not null, foreign key (referenced table `Class` column `id_class`)

Subject_id_subject: INT, primary key, not null, foreign key (referenced table `Subject` column `id_subject`);



Пользовательские роли

В данном проекте задумываются 2 пользовательские роли: ученик и учитель.

Для каждой из них будут реализованы следующие функции:

Админ: 1) добавление учеников

2) добавление предметов

3) добавление учителей

4) добавление классов

5) перераспределение вышеперечисленного

Ученик: 1) выбор предмета

2) выбор промежутка времени

3) просмотр оценок

4) просмотр посещаемости

Учитель: 1) выбор предмета

- 2) выбор промежутка времени
- 3) выбор класса
- 4) просмотр оценок
- 5) просмотр посещаемости
- 6) возможность внесения изменений в графе оценок/посещаемости

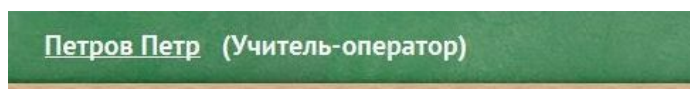
Т.е. различием между ролями будет возможность изменения/добавления данных.

UI/API

User Interface

1. Страница авторизации - два поля для ввода: логин и пароль. при нажатии на кнопку вызывается метод авторизации на сервере. При успешном ответе открывается страница журнала;
2. Страница журнала - при входе в качестве ученика появляется возможность выбрать предмет и период, за который пользователь хочет посмотреть оценки. При входе в качестве преподавателя добавляется возможность выбрать класс, в котором пользователь хочет посмотреть оценки. У преподавателя есть возможность ставить ученикам оценки, менять их и отмечать присутствие.

И для ученика, и для преподавателя выводятся оценки всего класса по выбранному предмету за выбранный период. Ориентировочный пример страницы журнала:



Журнал учителя

Параллель Класс Предмет Месяц

10 a Алгебра Октябрь 2012

Нельзя редактировать оценки более чем за 31 день назад.

№	Ученики	Дата	04	08	11	15	18	22	25	29		
		тип										
1.	Алексеев А. И.				3	н				4		
2.	Борисов Б. Б.				3	3	5					
3.	Владимиров В. И.			4	н			4	5			
4.	Григорьева Г. Г.	4			4		5	4	4			
5.	Дмитриев Д. Д.			5		н		5	н			
6.	Иванова И. И.	5	4	5	5							
7.	Константинов К. К.			5	5	5		4				
8.	Леонидова Е. Л.					4	3	5	4			
9.	Маринин М. М.	5	4	4		5	5	5				
10.	Николаев Н. Н.	4		4			3	4	3			
11.	Олегов О. О.	5	4		4	4		н				
12.	Петров П. П.		4		4			4	4			
13.	Романов Р. Р.			5		4	5		4			
14.	Сидоров С. С.		4	3		3						
15.	Тимурова Т. Т.	3			4							

3. Страница администратора - показывается таблица со всеми классами, в которой можно изменять их состав, список учителей и предметов. Также можно будет изменять для учителей список их предметов и классов.

API

- Ввод login/password и проверка совпадения введенных значений с базой данных на сервере;
- После получения id_user проверяем, является ли пользователь учителем (isTeacher);
- По значению id_user можем получить информацию о классе в котором учится ученик (или классы, в которых проводит занятия учитель);
- По id_class можем получить информацию о всех учениках в этом классе, и, соответственно, предметы, которые изучаются в этом классе (для учителей только присущие им предметы);
- Наличие информации о предметах и классе предоставляет доступ к информации о расписании (датах проведения занятий по выбранному предмету в выбранном классе);
- Наличие информации о расписании и ученике предоставляет доступ к информации об оценках;

Языки программирования

1 JavaScript

2 Html

3 Css

4 SQL

Субд

1 Реляционная mysql