

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ СОФИЯ –
ФИЛИАЛ ПЛОВДИВ

ФАКУЛТЕТ ПО ЕЛЕКТРОНИКА И АВТОМАТИКА

ДИПЛОМНА РАБОТА

Тема:

Уеб-базирано приложение за обработка и съхранение на
данни за учителска и ученическа дейност в учебно
заведение

Разработил: Георги Стефанов Георгиев

Фак.№:357749

Специалност: Компютърни системи и технологии

Образователно-квалификационна степен: Бакалавър

Научен ръководител: доц. д-р Ивайло Атанасов

Пловдив: 2016г.

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ, ФИЛИАЛ ПЛОВДИВ
ФАКУЛТЕТ ПО ЕЛЕКТРОНИКА И АВТОМАТИКА

Катедра: КСТ

Дата на задаване: Юни 2016

Дата на предаване: Септември 2016

Образователна степен: Бакалавър; Специалност: КСТ

Утвърждавам:.....

Декан:.....

(проф. д-р Гриша Спасов)

ЗАДАНИЕ
ЗА ДИПЛОМНА РАБОТА

на студента: Георги Стефанов Георгиев , фак. № 357749

Тема: Уеб-базирано приложение за обработка и съхранение на данни за учителска и ученическа дейност в учебно заведение

1. Описание на конкретната инженерна задача и вида на крайните резултати, които трябва да се решат в оригиналната част от дипломната работа: Приложението дава възможност за регистрация на три категории потребители: преподаватели, ученици и родители. Всеки от тях разполага с възможност да извършва справка за конкретен ученик и/или група от такива, в зависимост от правата, които притежава. Преподавателите могат да въвеждат оценки, присъствия и бележки само на ученици, на които преподават. Учениците могат да виждат, както своите оценки и присъствия, така и на други съученици, при определени условия. Възможност за споделяне и коментиране на домашни работи по някой от предметите.

2. Изходни цифрови данни за изчислителната част/ Функционални изисквания към софтуерната част: Функциите които трябва да поддържа системата са:

- въвеждане и редактиране на данни за учители, ученици и родители;
- редактиране на данните
- справки за състоянието на учениците

3. Изходни литературни и други източници:

Документация за PHP, MySQL, Apache Server, CSS, HTML, JavaScript

4. Съдържание на дипломната работа

- 4.1. Заглавна страница по образец
- 4.2. Оригинал на завереното дипломно задание
- 4.3. Съдържание
- 4.4. Увод
- 4.5. Постановка на проблема
- 4.6. Описание на средата за разработване
- 4.7. Програмна реализация
- 4.8. Изводи и резултати
- 4.9. Използвана литература
- 4.10. Приложения

Консултант:.....

(.....)

Научен ръководител:.....

(доц. д-р Ивайло Атанасов)

Студент:.....

Ръководител катедра:.....

(доц. д-р Николай Каканаков)

ДЕКЛАРАЦИЯ ЗА ОРИГИНАЛНОСТ НА ДИПЛОМНАТА РАБОТА

Долуподписаният, студент в специалност на ФЕА, ТУ - София, Филиал Пловдив, дипломиращ се през 20xx/xx учебна година, фак. №,

декларирам, че изложеното по изпълнението на конкретните задачи в дипломната ми работа на тема:

.....
.....

с ръководител:
в обем страници текст и страници Приложения; включително: брой фигури:
брой таблици:, е резултат от собствена работа.

Подпис:

дата:

МНЕНИЕ НА ДИПЛОМНИЯ РЪКОВОДИТЕЛ ЗА РАБОТАТА НА ДИПЛОМАНТА

Дипломната работа е изпълнена според заданието в пълен обем / в обем % (правилното се подчертава, ако изпълнението е под 100% при допускане до защита задължително се попълва мотивация за допускане)

и може / не може да бъде допусната до защита.

Мотивация:
.....
.....
.....
.....

Дипломен ръководител:
(.....)

Избран рецензент:

Ръководител катедра :
(.....)

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ, ФИЛИАЛ ПЛОВДИВ

ФАКУЛТЕТ ЕЛЕКТРОНИКА И АВТОМАТИКА

Катедра:
Образователна степен:

Утвърждавам.....
(Председател на ДИК:)

РЕЦЕНЗИЯ НА ДИПЛОМНАТА РАБОТА

на студента:, фак. №....., спец.....

на тема:

1. Описание на зададената инженерна задача и исканите крайни резултати:.....

2. Характер на дипломната работа:.....

3. Актуалност на разглеждания проблем:.....

4. Трудност на поставената задача:.....

5. Обем на дипломната работа: брой страници: брой фигури: брой таблици:

6. Оформление: форма и шрифт: номерация: качество на чертежите:.....
общо оформление:

7. Обща и техническа грамотност: правопис: стил:

владее на терминологията: коректност на означенията:

обоснованост на решенията и изводите:.....

8. В каква степен абсолвентът е проявил познаване на общите и специални проблеми на специалността:

9. В каква степен е решена поставената конкретна инженерна задача:.....

10. Доколко са съвременни използваните методи на решение:

11. Кой от претенциите са основателни и има ли в нея необявени оригинални резултати:

13. Оценка на увода и на работата с литературните източници:

14. Оценка на глава 3 - апаратна и/или софтуерна част и експеримент:

15. Оценка на глава 4 - изчислителна част:

16. Оценка на глава 5 -- приложимост на дипломната работа

17. Оценка на глава 6: икономическа оценка на резултатите от дипломната работа и оценка на техническата ефективност.....

18. Оценка на глава 7 - изводи :.....

19. Основни достоинства на дипломната работа:.....

20. Забележки, недостатъци и допуснати грешки в дипломната работа:

21. Предложение за допускане до защита и оценка:

Рецензент:

Дата:.....

(.....)

Съдържание

1.Увод	страница 4
2.Постановка на проблема.....	страница 5
2.1.Система за вход и оторизация на различните типове потребители.....	страница 5
2.2.Ограничения и възможности на различните категории потребители налагани от предметната област.....	страница 5
3.Описание на средата за разработване.....	страница 6
3.1.Избор на архитектура за изграждане на системата.....	страница 6
3.1.1.Еднослойна и двуслойна архитектури.....	страница 6
3.1.2. Трислойни и n-слойни архитектури.....	страница 6
3.1.3.MVC архитектура.....	страница 7
3.2.Използвани технологии и средства за програмиране.....	страница 8
3.2.1.Bootstrap	страница 8
3.2.2.PHP	страница 8
3.2.3.SQL.....	страница 10
3.2.4.Интегрирана среда за разработка – PhpStorm.....	страница 10
3.3.Проектиране и реализация на база от данни	страница 11
3.3.1.Процес на проектиране. Концептуален модел.....	страница 11
3.3.2.Проектиране и реализация на релационната база от данни.....	страница 11
Релационни модели на данните. Нормализирани релации.	страница 14
3.3.3.Физически модел на базата данни	страница 18
3.3.4.Структура и съдържание на таблиците в базата данни.....	страница 18
3.3.5.Типове данни използвани в дипломната работа.....	страница 18
3.3.6.Нагледно представяне	страница 19
3.4. Модули на приложението. Потребителски интерфейс.	страница 23
3.4.1.Административен модул.....	страница 23
3.4.2.Учителски модул.....	страница 24
3.4.3.Ученически модул.....	страница 26
3.4.4.Родителски модул.....	страница 26
4.Програмна реализация.	страница 27
4.1.Описание на файловата структура.....	страница 29
4.1.1.Директория content	страница 31
4.1.2.Директория controller	страница 31
4.1.3.Директория view	страница 32
5.Извод.....	страница 33
6.Използвана литература.....	страница 33

1.Увод

Дипломната работа има за цел да доразвие познанията на студента придобити по

време на следването. В същото време предоставя възможността тези познания да се концентрират в конкретна област. Област, която би могла да бъде максимално полезна на студента за неговата бъдеща реализация. Поради тази причина се спрях на тема, Уеб-базирано приложение за обработка и съхранение на данни за учителска и ученическа дейност в учебно заведение, използвайки скриптовият език за програмиране PHP и многопотребителската SQL система за управление на бази данни, на кратко MySQL. Тема ,която обхваща широк набор от проблеми и в същото време предоставя възможност за предприемане на различни подходи за тяхното решаване.

Приложението в разработената дипломна работа представлява ученическа информационна система, която разполага с възможност за регистрация на четири отделни категории потребители(Администратори, учители ученици и родители). Всяка от посочените категория разполага с различни възможности и предоставя различен тип информация. Основната идея е родители, ученици и учители да имат постоянен достъп до информацията, която ги касае свързана с училището.

бстр.

2.Постановка на проблема

С оглед резултатите от предварителното проучване и функционалността, която трябва

да предоставя системата, имаме наличие няколко задача(проблема), за решаване.

2.1.Система за вход и оторизация на различните типове потребители.

В случая имаме наличие на четири различни категории потребители. Това са:

- потребители занимаващи се с административната част на учебното заведение – условно наречени администратори. В случай възможностите, които ще имат са: одобрение на новорегистрирани потребители, което е част от сигурността за достъп до приложението;
- потребители със статут на ученици;
- учители;
- родители;

2.2.Ограничения и възможности на различните категории потребители налагани от предметната област.

Всяка от изброените по горе категории потребители притежава собствен профил, чиято информация в последствие може да бъде редактирана.

- Администраторите могат да въвеждат нови учебни дисциплини, да създават класове и учебни периоди по срок и година
- учениците от своя страна имат достъп до информация за своите съученици, след като са избрали класът, в който се обучават. Могат също да проверяват, както своите така и на съучениците си оценки по избран предмет.
- На учителите е предоставена възможността да оценяват ученици, като това може да се случва само на такива, на които преподават. При регистрацията си или на по късен етап отбелязват учебните дисциплини и класове, което от своя страна им позволява да изберат дали да бъдат класен ръководител, на кого и разбира се да отбележат отговорник на този клас. Могат да създават новини, статии, събития, които са достъпни до всички посетители на сайта.
- Родителите получават автоматично информацията за оценките на децата, добавени в техния списък.

3.Описание на средата за разработване

3.1.Избор на архитектура за изграждане на системата.

3.1.1.Еднослойна и двуслойна архитектури.

Всяко приложение, което работи самостоятелно и не разчита на други програми, може

да се каже, че е еднослойно. Днес много малко модерни приложения са изградени на този принцип. Голяма част от локалните приложения са базирани на двуслойната архитектура. Тези приложения често са наричани клиент-сървър приложения, тъй като в логиката на работа са заложени два основни обекта:

- клиенти – локалните приложения;
- сървъри – компютри или приложения, които обработват заявки на клиент-приложенията;

В тази ситуация, като правило, броят клиенти многократно надвишава броя сървъри.

Примери за приложения базирани на двуслойната архитектура са уеб-браузърите, e-mail клиентите, чат програмите и т.н.

Основната идея зад двуслойната архитектура е централизиране на достъпа до данните, докато в същото време потребителският интерфейс и бизнес логиката са съсредоточени в клиентските машини или приложения. При този подход логическата реализация на бизнес процесите и достъпът до данните са вградени в потребителския интерфейс (например процедури за работа с бази данни). Най-сериозното ограничение при реализиране на тази архитектура е, че клиентските машини за да поемат натоварването по обработката на данните, е необходимо тези данни да достигнат от сървъра до тях. В резултат на това, производителността намалява поради върхови натоварвания в мрежата.

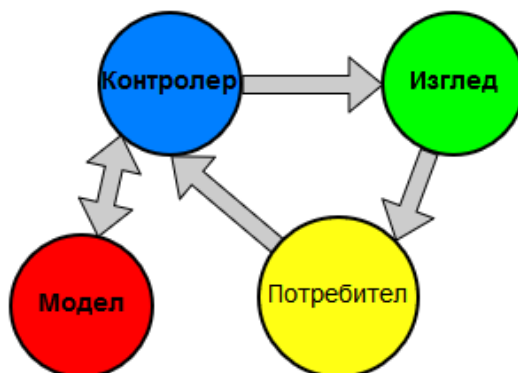
3.1.2. Трислойни и n-слойни архитектури.

При трислойната архитектура процесите свързани с управлението на бизнес логиката на приложението са отделени от потребителския интерфейс. Това създава допълнителен слой на физическа абстракция – логически слой (също наричан междинен слой). С негова помощ се обработват заявките към базата данни и информацията, която трябва да бъде представена чрез интерфейса на програмата, който се явява презентационен слой.

В зависимост от изискванията на даден проект, всеки слой може да бъде допълнително разделен на отделни абстрактни части, като най-често това се прилага върху логическия слой, в който по подразбиране се извършва обработката на най-голямото количество информация. С помощта на технологии за разпределени компоненти е възможно да разделим този слой между допълнителни машини в зависимост от логическото групиране на компонентите на приложението. В резултат се получава архитектура, която дефинираме като многослойна или n-слойна архитектура.

3.1.3.MVC архитектура.

Модел-Изглед-Контролер (Model-View-Controller или MVC) е архитектурен шаблон за дизайн, основан на разделянето на бизнес логиката от графичния интерфейс и данните в



фиг.1.Модел-Изглед-Контролер

дадено приложение.

Модел е ядрото на приложението, предопределено от областта, за която се разработва. В разработваното приложение това са класовете, описващи учебните групи, профили, събития, известия и др. В модела взлизат и операциите, които се извършват над тези обекти.

Изглед е тази част от изходния код на приложението, отговорна за показването на данните от модела. Например изгледът може да се състои от PHP шаблонни класове, JSP страници, ASP страници. Зависи от това какъв графичен интерфейс се прави и каква платформа се използва.

Контролер от своя страна е тази част от сорс кода(клас или библиотека), която взима данните от модела или извиква допълнителни методи върху модела, предварително обработва данните, и чак след това ги дава на изгледа.

Предимствата на MVC архитектурата са в това, че моделът е независим от контролера и изгледа.Той може да бъде планиран и осъществен независимо от другите части на системата. За един и същи модел могат да бъдат осъществени различни изгледи (интерфейси). Контролерът и изгледът могат да бъдат променени, без да се налага промяна в модела. Като недостатък можем да отбележим новото ниво на сложност, което системата достига. Това я прави по-трудно разбираема за работещите по нея.

3.2.Изпозвани технологии и средства за програмиране.

3.2.1.Bootstrap

Bootstrap е платформа, която е комбинация от HTML, CSS и JavaScript код, създадени от екипа на Twitter. Напълно поддържа CSS3 и HTML5. Това е важно, тъй като CSS3 и HTML5 са се превърнали в стандарт за уеб дизайн през последните години. Съвместимостта с всички основни браузъри го прави предпочитан от голяма част от програмистите. Сайтовете изградени с Bootstrap изглеждат зашеметяващо и работят правилно независимо от устройството и/или браузърът, които посетителите използват. С помощта на комбинация от JavaScript, CSS и интелигентни мрежи, Bootstrap позволява да се изпълни почти всеки елемент от дизайна, който може да се изиска. Предимствата на този фреймуърк са, че е:

- бърз;
- напълно приспособим;
- лесен, елегантен и интуитивен;
- отзивчив от самото началото;
- притежава перфектна grid система;
- обширен списък на компонентите;
- пакетни Javascript плъгини;
- притежава всички елементи, които са необходими за разработването на уеб сайт - таблици, форми, бутони, падащи менюта и т.н.
- предоставя стилове за всички основни HTML елементи;

Bootstrap екипа постоянно актуализира рамката. Всички изброени по-горе характеристики означават много перфектно проектирани шаблони. Bootstrap шаблоните са лесни за поддръжка и не е необходимо да бъдете уеб разработчик за да управлявате вашия уеб сайт.

3.2.2.PHP

PHP е сървърен скриптов език, проектиран специално за създаване на интернет базирани приложения. PHP кодът може да се вгради в HTML страница и ще се изпълнява при всяко нейно зареждане. Кодът се интерпретира от уеб-сървъра и генерира изходни данни, които посетителят на сайта или потребителя на уеб-приложението вижда. Основните фактори, които дават предимство на PHP пред другите езици за уеб-програмиране са следните:

- **Производителност.** PHP значително надминава по производителност своите конкуренти. Той се справя отлично с обслужване на много посещения едновременно дори хардуерът на сървъра да не е достатъчно бърз.
- **Интеграция с база данни.** PHP притежава интерфейс за много различни системи бази данни.
- **Преносимост.** PHP може да работи еднакво върху UNIX-базирани операционни системи и върху Windows.
- **Вградени библиотеки.** PHP е проектиран специално за работа с уеб и за целта притежава голямо количество вградени библиотеки и функции за решаването на стандартни задачи.
- **Отворен код.** Потребителят може сам да добавя нови функционалности на езика. Самият продукт е безплатен.

PHP може да бъде интерпретира като всеки друг CGI скриптов език или като Apache модул. Apache е достъпен и за Windows платформа и PHP се поддържа за такава комбинация. Съществуват и множество други уеб сървъри, които могат да се използват с PHP, но само чрез Apache сървър PHP може да бъде компилиран като модул, останалите са инсталации на CGI интерпретатор.

Когато PHP е компилиран като Apache модул той се изпълнява в същото адресно пространство като това на процеса на уеб-сървъра, което осигурява значително по-добро време за интерпретация на скрипта. Определени характеристики, като например постоянна връзка с база данни (persistent database connection) са налице само във версията на Apache модул. Когато PHP е конфигуриран като CGI интерпретатор, всеки път когато PHP скриптът се интерпретира, уеб-сървърът създава инстанция на PHP интерпретатора, която интерпретира скрипта. Това се отразява на производителността, но от друга страна, дава и определени предимства. CGI интерпретатора може да се изпълнява под различни потребителски идентификатори (потребители). CGI версията може да бъде инсталирана на уеб сървъри от потребители, които нямат администраторски права. Друго предимство е, че е много по-трудно за “вредни” или недобре написани скриптове да доведат до срыв на уеб-сървъра. Динамичните интернет страници най-често осъществяват достъп до база данни.

3.2.3.SQL

Език за структурирани запитвания (Structured Query Language - SQL) е универсален компютърен език използван,при създаване, модифициране и управление на данни в релационни бази данни. SQL е преди всичко информационно-логически език, предназначен за описание, извличане и модификация на съхранявани данни.

Основни типове данни, с които работи SQL и които се съхраняват в релационните бази данни са:

- Числа (integer, real, floating point)
- Низове (string, char)
- Дати и време (date, time)
- Двоични (Binary)
- Големи текстове (BLOB)
- null стойности

Днес SQL, в една или друга степен, се поддържа от всички релационни системи за управление на бази данни (СУБД). СУБД използвана от разглежданата система е MySQL. Широко се използва при изграждането на уеб сайтове, особено такива с по-голям обем информация, изискваща бързо четене и обработка.

3.2.4.Интегрирана среда за разработка – PhpStorm.

JetBrains PhpStorm е интегрирана среда за разработка (IDE) с платен лиценз и поради тази причина за създаването на дипломната работа ще използвам 30-дневния пробен период. IDE-то е предназначено специално за създателите на уеб приложения с помощта на PHP, HTML, CSS, JavaScript и XML.

PhpStorm помага на разработчиците да създават и модифицират сорс код, независимо какъв език за програмиране използват. Освен базови функции платформата предлага и по-развити възможности от рода на макроси, анализ на кода и бърза навигация.

Средата се отличава с интуитивен интерфейс и добре организирани менюта. Налични са възможности за редактиране на Run/Debug конфигурациите, както и за промяна на опциите на интерпретатора.

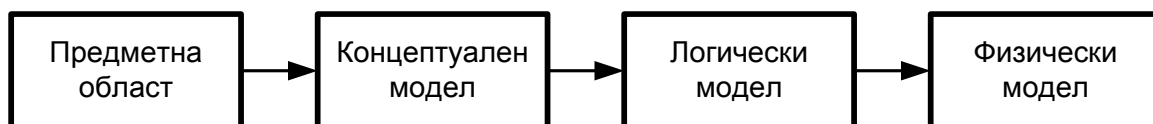
3.3.Проектиране и реализация на база от данни

Базите от данни съдържат организирана по специален начин информация. Всяка БД е предназначена за информационно осигуряване на определен кръг от потребители. В зависимост от потребността на потребителите се определя предметната област за БД. Целта е да се определи съвкупността от обекти, характеризиращи информационното пространство, което е нееднородно, защото съдържа информационни обекти. Предметната област се състои от множеството обекти, от действителността заедно с техните свойства, взаимоотношения и процеси на взаимодействие между тях.

3.3.1.Процес на проектиране. Концептуален модел

Проектирането на база от данни преминава през три етапа:

- проектиране на концептуален модел – представяне на всички обекти и взаимовръзките между тях;
- проектиране на логически модел – представяне на обектите и взаимовръзките с помощта на моделите на данни: мрежов, йерархичен или релационен; представяне на графически модели, изхождайки от модела на данни;
- проектиране на физически модел – методите за съхраняване и достъп до данните, а също и методите за достъп до ОС;

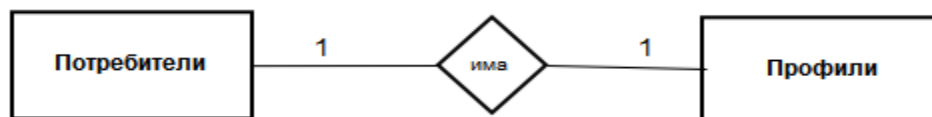


Един от най-популярните концептуални модели на данни е моделът „Същност-Връзка”, накратко наричан ER-модел (Entity-Relationship). Моделът е предложен от Питър Чен (Chen) през 1976 г. Моделирането на предметната област се базира на използване на графични диаграми, в които обектите и връзките между тях се представят с графични символи. Моделът ER е доразвит от множество разработчици и днес съществуват редица негови разновидности: модел на ORACLE; модел на Vantage Team Builder; модел на Баркер; метод IDEF1, разработен от Т. Ремей (T.Ramey), и неговата доразвита версия IDEF1X (използва се в ERwin, Design/IDEF и др.); метод IE (Information Engineering) и т. н. Разликите между отделните модели се състоят в графичното изобразяване на обектите, атрибутите и връзките, както и в степента на отразяването на особеностите на обектите, атрибутите и връзките.

В приложение 1. имаме графично представяне на базата от данни, чрез „ER диаграма“, предмет на разработената дипломна работа, свързана с дейността на учебно заведение. Компонентите, които съдържа тази а и всяка ER-диаграма са:

- обект – наименованието на реален обект от предметната област. В случая имаме потребител, профил, ученик, учител, родител, предмет, клас, адрес и събития.
- клас обекти ще наричаме множество от обекти, които имат сходни свойства. В ER диаграмата са изобразени с правоъгълници. Учители, ученици, родители, потребители, профили, предмети, класове и събития;
- атрибути се наричат свойствата на всеки обект в даден клас. Изобразяват се като елипси. Атрибутите, които имаме в класа обект потребители са потребителско име, парола, електронна поща, дата на регистрацията и статус. Всеки обект си има свой собствен уникален “ключ“, което го идентифицира спрямо другите обекти в класа. В класа Профили например можем да допуснем, че атрибута ЕГН се явява “ключ“. По принцип ЕГН-то трябва да е индивидуално за всеки човек, въпреки че това не е съвсем така.
- връзки – използват се за свързване на различни обекти от един или различни класове. Характеризират се със свойства и също характеризират обектите. Изобразяват се с ромб на ER-диаграмата и се разделят на три основни типа:

Връзка 1:1. Един обект от даден клас е свързан точно с един обект от друг клас.



фиг.2. Връзка 1:1

Връзка 1:M(едно към много) се получава когато един обект от даден клас може да има повече от една връзка с обекти от друг клас, но обект от другия клас може да има само една връзка с обект от дадения.



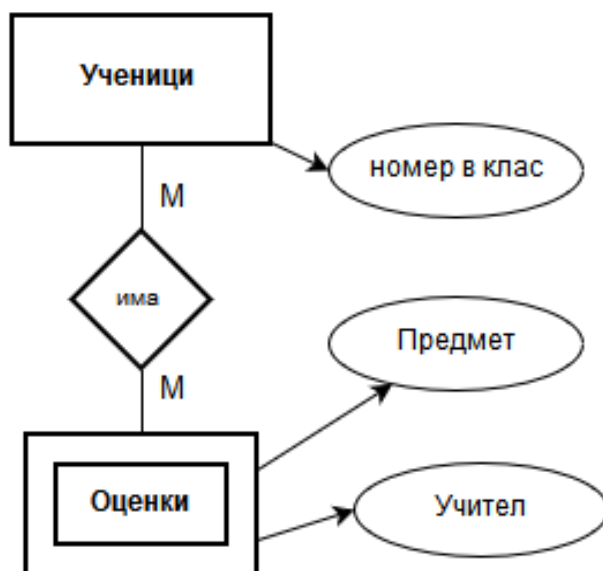
фиг.3. Връзка 1:M

Връзка М:М(много към много) се получава когато обект от даден клас има повече от една връзка с обекти от друг клас и обект от другия клас има повече от една връзка с обектите от дадения.



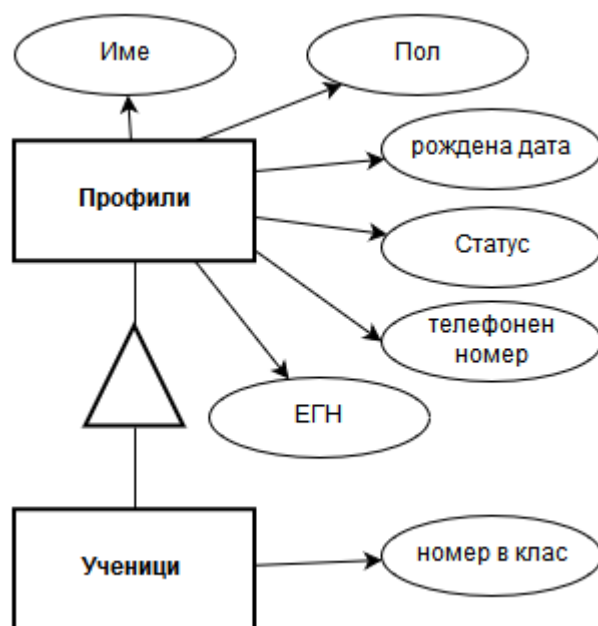
фиг.4.Връзка М:М

- характеризиращ обект – обекти от предметната област, които не могат да съществуват самостоятелно или иначе казано не можем да имаме оценка без да бъде посочен преподавател и съответно предмет.



фиг.5.Характеризиращ обект

- подклас обекти – това са обекти, който притежават свойства на даден базов клас обекти, но имат и допълнително свои атрибути. Тази връзка между базов клас и негов подклас се изобразяват с триъгълник.

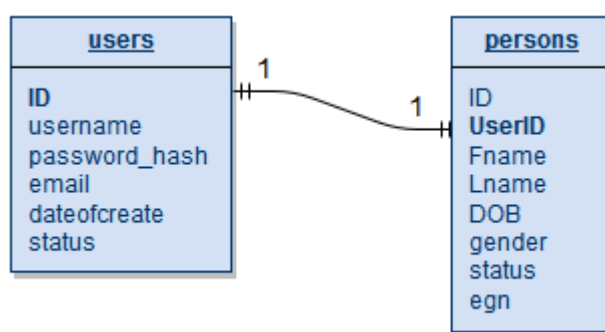


фиг.6.Подклас обект

3.3.2.Проектиране и реализация на релационната база от данни.Релационни модели на данните. Нормализирани релации.

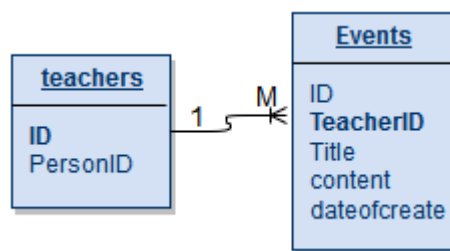
Формиране на схема на релационната БД (списък от схеми на отношенията). Изследват се взаимодействията между обектите, които се описват чрез БД и се проектира релационният модел:

- "един към един" (1:1) – между обектите А и В съществува такава връзка, когато на даден екземпляр от А съответства само един екземпляр от В и обратно;



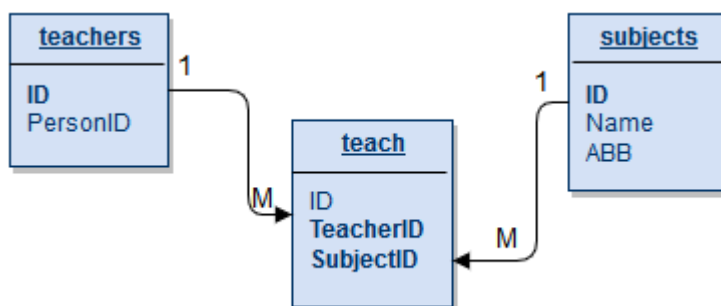
фиг.7.Релационна връзка 1:1

- "един към много"(1:M) – между обектите А и В съществува такава връзка, когато на един екземпляр от А, съответстват един или много екземпляри от В, а на всеки екземпляр от В съответства само един от А;



фиг.8.Релационна връзка 1:M

- "много към много"(M:M) – между A и B съществува такава връзка, когато на екземпляр от A, съответстват един или много екземпляри от B, а на екземпляр от B съответстват един



фиг.9.Релационна връзка M:M

или много от A

Релационните бази данни ни позволяват да съхраняваме числа, дати и символни низове. Много от тях ни позволяват да съхраняваме и големи бинарни обекти, като звук или изображения. Релационните бази данни се използват, когато обектите имат много данни и малко връзки помежду си. Повечето мощни системи за управление на бази данни използват така наречения релационен модел данни. Релационният модел се характеризира с една или повече връзки между различни таблици. В основата на релационната теория за представяне на данните лежи понятието двумерна таблица. Двумерната таблица е един от най-простите начини за представяне на данните от гледна точка на потребителя. Всяко представяне на данните макар и с известно дублиране на информацията може да се представи с двумерна таблица.

Всяка таблица притежава следните свойства:

- всеки елемент от таблицата представлява един елемент от данни (атрибути); отсъстват повтарящите се групи данни
- всички стълбове от таблицата са еднородни, т. е. елементите в даден стълб имат една и съща природа, еднаква същност на стълбовете се присвояват имена, които са уникални

- в таблицата няма два еднакви реда
- при разглеждане на операции с такава таблица, нейните редове и стълбове могат да се разглеждат в произволна последователност и ред без да се прави връзка с тяхното информационно съдържание и смисъл
- таблица от този тип е прието да се нарича релация (отношение)

Описанието на всяка релация се състои от име на релация, след което в малки скоби следва списък от атрибутите ѝ. Това описание на релацията се нарича схема на релацията. Съвкупността от такива схеми, които се използват за представяне на информацията за обектите от дадена предметна област, се нарича схема на релационната БД.

Наборът от означения на елементите от данни от един тип, т. е. един стълб от таблицата се нарича домен. Съвкупността означения на домените в един ред е прието да се нарича кортеж.

Всяка релация има първичен ключ и притежава най-малко един възможен ключ, тъй като най-малко комбинацията от всички атрибути на релацията удовлетворява условието за уникалност. Един от възможните ключове на дадена релация се избира първичен. Останалите възможни ключове, ако има такива, се наричат алтернативни.

Външният ключ е атрибут или комбинация от атрибути на дадена релация R2, значението на който задължително трябва да съвпада със значението на първичния ключ в някаква друга релация R1.

На основата на тези правила се формират две правила за цялост на релационния модел на данните:

- Цялост по същности: Не се допуска който и да е атрибут, участващ в първичния ключ на дадена базова релация, да приема неопределено значение
- Цялост на връзките: Ако релацията R2 включва някакъв външен ключ FK, съответстващ на първичния ключ PK на друга релация R1, то всяко значение на FK в R2 трябва или:
 - да бъде равно на значение на PK в някои кортежи на R1
 - да бъде напълно неопределено, т.е. всяко значение на атрибута, участващ в това значение на FK да бъде неопределено.

Целостта на БД предполага изпълнението на други ограничения:

- Стойностите на даден атрибут на релацията да се намират в определен интервал, т.е. да принадлежат на точно определено множество от стойности.
- Даден атрибут, който не е ключ, да не може да приема неопределени (null) стойности.

Дизайнерите на бази данни говорят за пет нива на нормализация – стандарти, осигуряващи съвместимостта на базата данни. Нормалните форми имат йерархия. Между атрибутите в една релация съществуват определени функционални зависимости. Някои от тях са нежелателни поради страничните ефекти или аномалии, които възникват при модификации на БД. Това налага да се прави декомпозиция на релациите, при което дадено множество от релации се заменя с друго множество от релации (проекция на първоначалното множество). Дефинирани са пет нормални форми:

- Първа нормална форма. На всяко сечение ред/колона трябва да има една и само една стойност.
- Втора нормална форма. Всяка неключова колона трябва да зависи от целия първичен ключ. Ако първичният ключ е комплексен, образуван от повече от един компонент, нито една от неключовите колони не трябва да оказва влияние на подмножеството на първичния ключ.
- Трета нормална форма. Нито една неключова колона не трябва да зависи от друго неключово поле.
- Четвърта нормална форма. Не трябва да има независими връзки един-към-много между колоните с първичен ключ и неключовите колони.
- Пета нормална форма. Таблиците са разделени във възможно най-малки части, за да се елиминира всеки излишък в таблицата. В най-крайния случай, таблиците в пета нормална форма могат да се състоят от първичен ключ и една неключова колона.

Процесът, насочен към преобразуване на релационната схема, при който се налагат известни ограничения, елиминиращи някои нежелателни свойства, се нарича нормализация на релационната схема.

3.3.3.Физически модел на базата данни

Проектирането на физическия модел на базата от данни става като се спазват резултатите от направеният анализ. На практика това става като се спазват следните стъпки, в които е взето предвид всичко гореспоменато:

- Определяне на целите на приложението: определят се всички данни, които ще обработва системата;
- Определяне на таблиците: данните се разпределят в таблици след изясняване на целите на приложението;
- Определяне на полетата на таблиците: тип на данните, обхват, стойност по подразбиране и др. ;
- Определяне на връзките между таблиците (взаимодействието между обектите): определят се връзките на данните от една таблица с данните от друга таблица и така за цялата база от данни. Ако е необходимо, се нормализира базата от данни.

3.3.4.Структура и съдържание на таблиците в базата данни.

Едно от най-големите предимства на една релационна база от данни е скоростта, с която тя може да сортира и извлича определена информация от масивите, които управлява. За да се постигне това, MySQL и всички останали системи за управление на бази от данни използват оптимизирани механизми за съхранение на данни, наречени индекси. Индексите позволяват на системата да създаде представяне на колона, която може да претърсва със значителна скорост. В дефиницията на една таблица колоната или колоните, които декларираме като първичен ключ, автоматично ще бъдат индексирани.

Имената на таблиците са добре подбрани, за по-голяма яснота. Имената на колоните в таблиците също достатъчно добре показват, каква информация съдържат. Типът на колоните е така избран, че да се въвеждат само такива данни, които отговарят на този тип, за да се избегнат нежелани грешки.

3.3.5.Типове данни използвани в дипломната работа.

В дипломната работа са използвани полета от следните типове:

- INT - цяло число в интервала [-2147483648, 2147483647] или [0, 4294967295] като unsigned. В дипломната работа са използвани за представяне на данни, представляващи цели числови стойности, Освен това почти всички полета, които формират PRIMARY KEY са от този тип.
- TEXT - Тези полетата работят както VARCHAR, но за разлика от него самите данни се съхраняват като обекти извън самата таблица, а в таблицата се пазят само референции към тях. Максимална дължина 65535 символа.
- VARCHAR – полета от тип символен низ с променлива дължина. Използва се за тип на полета, в които ще се съхраняват символни низове, като например имена, описания, 20стр.

телефонни номера и т.н.

- DATETIME – поле от вида “YYYY-MM-DD HH:MM:SS”. Използват се за съхранение на информация за дата и час.
- DATE – Дата във формат YYYY-MM-DD. Минималната стойност е ‘1000-01-01’, а максималната ‘9999-12-31’.
- ENUM – полетата от този тип съхраняват данните във вид на символен низ, но особеното при тях е това, че могат да приемат само стойности от предварително дефинирано множество, което се дефинира при създаването на таблицата и колоната. Полета от този тип в дипломната работа са статуса на потребителите, както и полето определящо пола.

3.3.6.Нагледно представяне

В *приложение 2* е представена пълната структура на релационната базата от данни, както и връзките между отделните таблици. Какво всъщност показва тази релационна схема? Как са създадени тези таблици? Какво означават? Какви са отношенията?

Всяка релация (таблица) в базата данни носи уникално име: persons, users, students, teachers,...и т.н.

Всеки атрибут носи уникално име в рамките на дадена релация.

Всяка релация съдържа уникални записи, не може да съдържа повтарящи се идентични записи. Това условие се гарантира от наличието на атрибут(ID) за всяка една релация, които нараства автоматично при нов запис.

Няма определен ред, в който се разполагат записите в дадена релация или атрибутите в даден запис.

Стойностите в записите са атомарни – те не могат да се състоят от различни типове данни (данни от различни домейни) или да са резултат от изчисление или конкатенация.

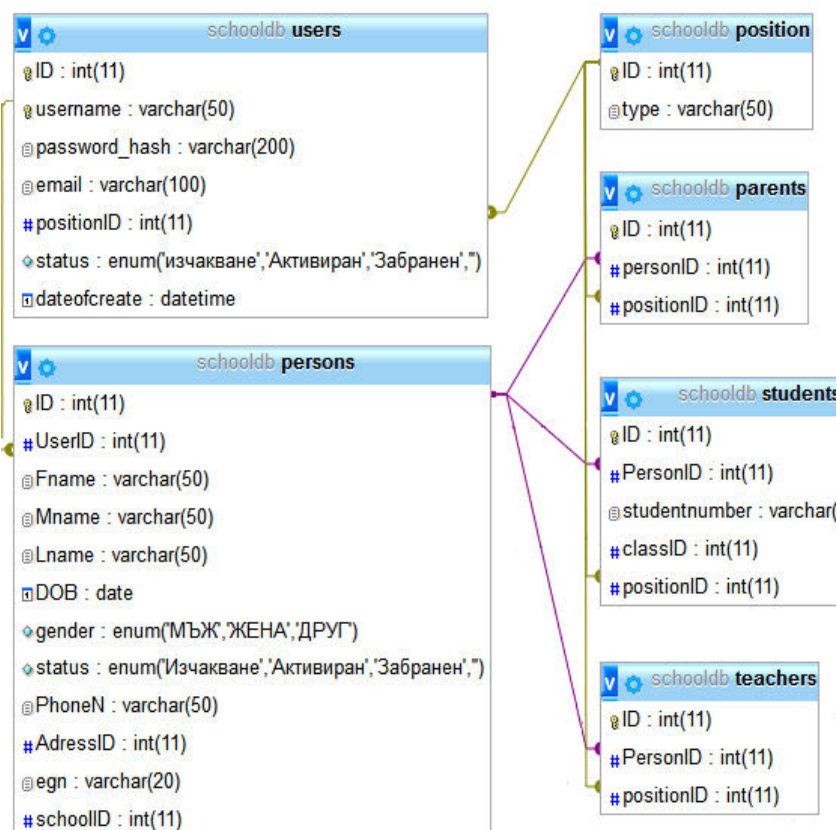
За създаването на таблиците използваме SQL заявки и конзолното меню на phpStorm.

Пример:

```
CREATE TABLE persons
(
  ID INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
  UserID INT(11) NOT NULL,
  Fname VARCHAR(50),
  Mname VARCHAR(50),
  Lname VARCHAR(50),
  DOB DATE NOT NULL,
  gender ENUM('МЪЖ', 'ЖЕНА', 'ДРУГ'),
  status ENUM('ИЗЧАКВАНЕ', 'АКТИВИРАН', 'ЗАБРАНЕН', ''),
  PhoneN VARCHAR(50),
  AdressID INT(11),
  egn VARCHAR(20) NOT NULL,
  schoolID INT(11),
  CONSTRAINT persons_ibfk_2 FOREIGN KEY (AdressID) REFERENCES address (ID),
  CONSTRAINT persons_ibfk_3 FOREIGN KEY (schoolID) REFERENCES schools (ID),
  CONSTRAINT persons_ibfk_4 FOREIGN KEY (UserID) REFERENCES users (ID)
);
CREATE INDEX AdressID ON persons (AdressID);
CREATE INDEX schoolID ON persons (schoolID);
CREATE INDEX UserID ON persons (UserID);
```

Дефинираме ID, като първичен ключ, който не може да бъде NULL и автоматично се увеличава при следващ запис. UserID не може да приема нулева стойност. Атрибутите Fname, Mname, Lname са предвидени за имена с не по-голяма от 50 символа дължина. DOB или рожденната дата е от тип DATE и не може да приема нулева стойност. Gender полето може да приеме само една от въведените му стойности („МЪЖ“, „ЖЕНА“, „ДРУГ“). Аналогичното следва и за status. Следват дефинирането на външните ключове(отношенията) между таблиците и създаването на индекси.

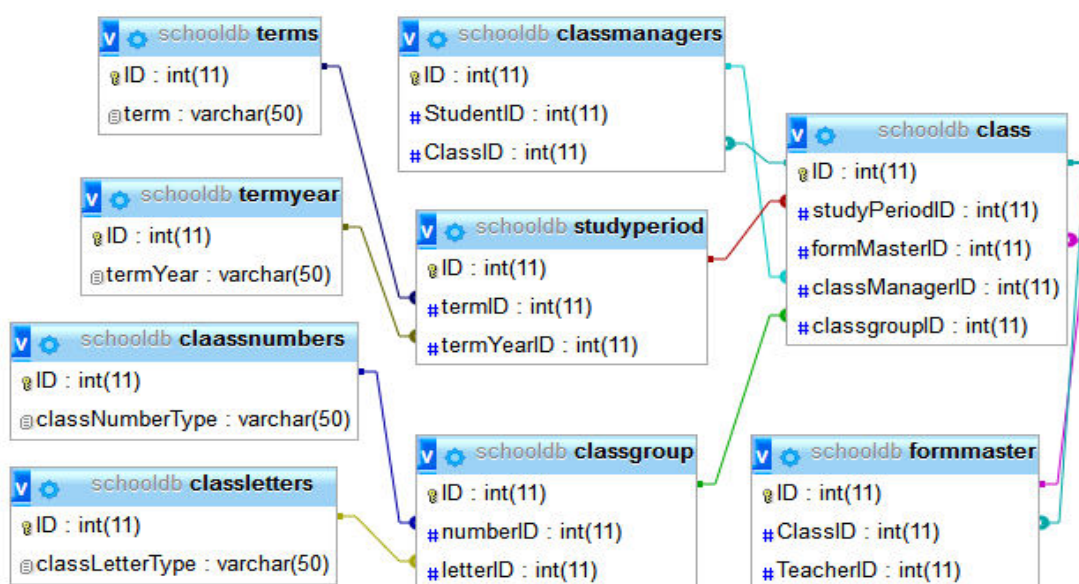
Предвид предметната област на избраната дипломна работа и с оглед на функционалността, която трябва да изпълнява приложението, дефинираме основен обект условно наречен – профил. Този обект от своя страна има три подобекта: учител, ученик и родител. Всеки профил се състои от производна релация, която се е получила след отношението между други релации, в случая между users и persons. Тези релации в приложенията за база данни се наричат заявки или изгледи.



фиг.10.релационна схема на всеки профил

Чрез трите подобекта се дефинират и другите информационни структури, които системата управлява – учебни групи, предмети, оценки, присъствия, класни ръководители и т.н.

Учебните групи или класовете са съвкупност от учебен период, наименование на групата, отговорник на класа както и класният ръководител.



фиг.11 Релационна схема(изглед) за всеки клас

В базата данни участват и такива таблици, с помощта на които лесно може да се изведе моментното състояние на даден подобект. Например:

- информация за предметите и/или класовете, по които даден учител преподава; Такива таблици са: teach, teaching
- информация за текущите оценки и присъствия на конкретен ученик – evalution, present;

3.4. Модули на приложението. Потребителски интерфейс.

Приложението се състои от четири модула:

- административен модул;
- учителски модул;
- ученически модул;
- родителски модул.

Началната страница на приложението е достъпна за всички потребители независимо от това дали имат или не регистрация в системата. Разполагат с възможността да разглеждат статий публикувани от учителите. За да има достъп до информацията в системата всеки потребител трябва да се е регистрирал преминавайки през три етапа.

За целта се използва стандартен модул за регистрация, които включва потребителско име, парола и електронна поща. На вторият етап попълва своята лична информация: три имена, рождена дата, егн, адрес, пол, телефон за връзка.

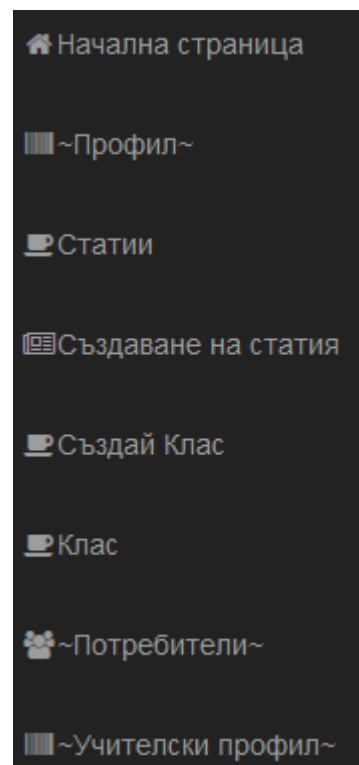
Крайният етап от регистрацията на всеки потребител се извършва от потребители с администраторски права.

3.4.1.Административен модул.

Административният модул се състои от два екрана. Единия съдържа основна информация за потребителя, докато вторият отговаря за одобрението на новорегистрираните потребители. Организацията по извеждането на информацията за всички потребители е възможна посредством три начина:

- извеждане на послени четири новорегистриране потребителя;
- възможност за търсене по ЕГН;
- извеждане на списък от всички потребители посредством страниране.

Освен одобрение, от администратора зависи, да бъде определен типа на всеки потребител. Дали ще бъде учител, ученик или нечий родител.



фиг.11. Меню за
учителски модул

Всички потребители										
NEXT										
Потреб.Име	Дата на създаване	Потреб.Статус	Три имена	Проф.Статут	Роден	Е-майл	ЕГН/лк.№	Позиция	Адрес	Пол
tretanova	2010-05-01 07:24:39	<input type="text"/>	dasdasdas dasdas	<input type="text"/>	2016-08-11	tretanova@bav.bg	asdas	<input type="text"/>	dsadasd rp.dasdasd улица:dasdas	МЪЖ
bart	2016-08-26 22:44:26	АКТИЕ <input type="text"/>	Барт Джей	АКТИ <input type="text"/>	1991-09-01	bart@example.com	910901	учен <input type="text"/>	България гр.Пловдив улица:Копринка №100	
dragan	2016-08-31 18:17:37	АКТИЕ <input type="text"/>	Драго Петков	АКТИ <input type="text"/>	2016-08-02	dragan@abv.bg	14416516	учен <input type="text"/>	България гр.Пловдив улица:Все тая №55	МЪЖ
userPesho	2016-09-01 15:24:29	АКТИЕ <input type="text"/>	Петър Димитров	АКТИ <input type="text"/>	2000-02-15	peshoeloser@xaxa.net	0102154525	учен <input type="text"/>	България гр.Благоевград улица:няма адрес	МЪЖ

фиг.12. Списък на всички потребители изведено, чрез стрнциране.

3.4.2.Учителски модул

Учителският модул се състои от седем екрана:

- -Профил – съдържащ основна информация за потребителя;
- -Статии – списък на всички статии и възможност за тяхната редакция и/или изтриване
- Създаване на статия – заглавие и съдържание;

Създай клас														
срок:	<input type="text"/>	учебна година:	<input type="text"/>	налични периоди	<input type="text"/>	Изпрати								
клас от 1 до 12	<input type="text"/>	подгрупа:	<input type="text"/>	налични класове:	<input type="text"/>	Изпрати								
въведи предмет:	<input type="text"/>	Абrevиатура на предмета:	<input type="text"/>											
налични предмети	<input type="text"/>	Изпрати												

фиг.13.Създай клас

- Създай клас – дава възможност за създаване на учебен период, клас, подклас(група), учебни предмети;
- Клас – извежда списък на учениците от даден клас, за конкретен учебен период;
- Потребители – списък на всички потребители в зависимост от типа им на регистрация;
- Учителски профил – възможност за редакция на лична информация. Добавяне/премахване на предмети по преподаване. Добавяне/премахване на класове.

- Назначаване на класен ръководител. Информация за отговорник на ръководеният клас;

Регистрирани сте в системата, като учител
потребителско име adam

Промени

Една
Първото Ви име е....

Крапабел
Врото Ви име.....

Крапабел
И третото :).....

Роден сте на ?.... 1954-11-24

5411244525
ЕГН или номер по лична карта

португалия
От коя държава сте?

варна
Град или село....

улица № 1
Адресът, на който живеете

088542524
Телефон за връзка.....

пол

Изпрати [\[Начало\]](#)

фиг.14.Модул за редактиране на лична информация

Преподавате на:

Клас	За периода	Изтрий
1 - а	първи - 2000/2001	изтрий
1 - г	първи - 2001/2002	изтрий
3 - а	първи - 2000/2001	изтрий

изберете клас

Добави

фиг.16.Модул за добавяне и премахване на клас

Предмети, по които преподавате

Предмет	Съкращение	Изтрий
Български език и литература	БЕЛ	изтрий
Математика	МАТ	изтрий
Околен свят	ПРЗ	изтрий
Компютърни науки	КМН	изтрий

налични предмети

Добави

фиг.15.Модул за добавяне и премахване на учебни предмети

3.4.3.Ученически модул

Ученическият модул разполага с три екрана

Профил – освен основната информация отговаряща за съответният ученик, може да се направи и справка за текущите оценки по избран предмет;

Потребители – списък на съучениците и справка за техните иценки по избран предмет.

Ученически профил - възможност за редакция на лична информация. Избиране на клас и период на обучение. Добавяне на профилна снимка.

Предмет	Оценка
Математика	7 от дата 2016-09-10 17:04:21
	10 от дата 2016-09-10 16:39:05
	8 от дата 2016-09-10 16:11:04
	10 от дата 2016-09-10 15:19:22
	2 от дата 2016-09-10 14:52:04

фиг.17.Модул за проверка на оценка

3.4.4.Родителски модул

Притежава два екрана.

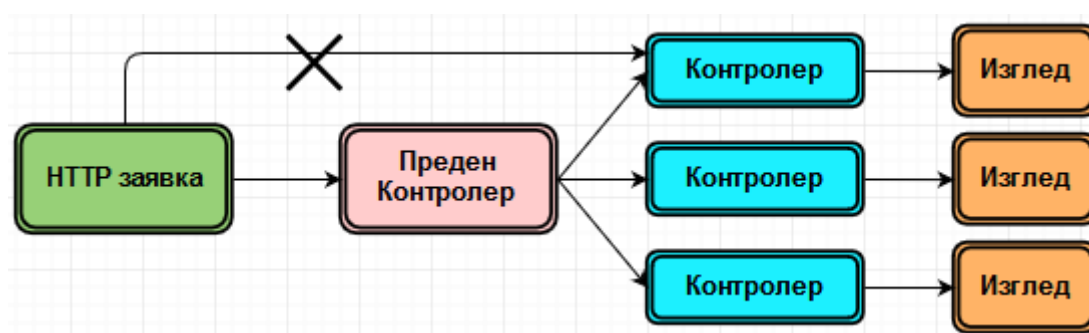
Профил - съдържащ основна информация за потребителя;

Родителски профил - възможност за редакция на лична информация. Търсене на ученици по ЕГН и добавянето на роднинска връзка. Справка за оценките по всички предмети.

4.Програмна реализация.

За написването на системата е използвана MVC архитектура, за програмен език PHP, а за система за управление на база данни е използван MySQL.

Предният контролер представлява единствена точка за достъп до едно уеб приложение (пр. index.php), което обработва всички заявки. Този код е отговорен за зареждането на всички зависимости, обработката на заявката и изпращането на на отговора обратно до брауъра. Предният контролер може да е полезен, защото насърчава разделянето на кода на модули и предоставя централно място за закачане на код, който трябва да се изпълнява с всяка заявка (като например изчистване на входните данни).



Фиг. 18 Преден контролер

Шаблонът Модел-изглед-контролер (MVC) позволява да се разбие кода на логически обекти, които вършат точно определена работа. Моделите служат, като слой за достъп, извличане и форматиране на данни в цялото приложение. Контролерите обработват заявките, обработват данните получени от моделите и зареждат изгледите за изпращане като отговор. Изгледите формират изходните данни по подходящ начин за визуализация.

4.1.Описание на файловата структура.

Програмните файлове на системата са организирани в отделни директории в зависимост от

своята същност. Пълна структура на файловата система е показана в *Приложение 3*.

.htaccess е конфигурационен файл и работи на ниво директория, т.е. може да промени част от глобалната конфигурация на сървъра, поставена от `php.ini` файла, за директорията, в която е поставен и нейните папки. Главната му цел е да определи кой има достъп до съдържанието на определена папка през Интернет. Използва се при Apache web базирани сървъри. В случая се използва за пренасочване на всички заявки, които не включват папката `content` към `index.php`.

30Стр.

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REQUEST_URI} !^(/[a-zA-Z0-9_-]+)?/content/.*$
RewriteRule ^ index.php
</IfModule>
```

`index.php` -

```
require_once('config.php');//извиква config.php файла
require_once('functions.php');//извиква functions.php файла
session_start();//стартира сесия
$requestParsed = parseRequest();//парсва заявката, разделя контролера ,
// действието , параметрите
processRequest($requestParsed);
```

`config.php` – Дефинирани са няколко константи.

```
define('APP_ROOT', '/school');
define('DEFAULT_CONTROLLER', 'home');
define('DEFAULT_ACTION', 'index');
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'schooldb');
```

`functions.php` – в този файл, парснатата заявка намира контролера, намира действието и му подава параметри ако има такива.

В инфраструктурата на архитектурата влизат и други два файла: *BaseModel.php* и *BaseController.php*.

- **BaseModel** – капсулира връзката с базата данни в променливата `$db`. Добавя енкодинг. Ако не може да се свърже с базата подава съобщение за това.
- **BaseController** – зарежда модела по подразбиране (`BaseModel.php`). Взима и изпълнява `view/_layout/header.php`, след което изпълнява резултата от текущото действие на контролера и накрая изпълнява `view/_layout/footer.php`. Съдържа и функции за:
 - препращане `redirectToUrl()` и `redirect()`;
 - проверка дали сме влезли в системата `authorize()`;
 - обработка процесите по валидиране на получените от формите, данни.

header.php - съдържа bootstrap библиотеки, скриптове за календар и основното меню на приложението. То се оформя според типа на регистрираният потребител.

```
<link href="<?=APP_ROOT?>/content/css/bootstrap.min.css" rel="stylesheet">
<link href="<?=APP_ROOT?>/content/css/datepicker.css" rel="stylesheet">
<script src="<?=APP_ROOT?>/content/js/jquery.js"></script>
```

Добавяне на bootstrap и календар към проекта.

31Стр.

Да разгледаме основните директории съставлящи проекта.

4.1.1.Директория content

Съдържа CSS файлове, шрифтове, скриптове и профилните снимки на потребителите.

datepicker.css – файл, които се използва за календар.

school-admin.css – допълнителен CSS.

bootstrap.min.css и *jquery-3.0.0.min.js* - конфигурационните файлове на Bootstrap

4.1.2.Директория controller

Контролерът управлява исканията на потребителите (получени като HTTP GET или POST заявки). Неговата основна функция е да извиква и координира необходимите ресурси / обектите, необходими за изпълнение на желаното от потребителя действие. Контролерът взима подходящия модел за изпълнение на задачата и след това избира правилния изглед за предоставянето на информацията.

ClassController.php – съдържа три основни действия

onInit()- проверява дали сме влезли в системата и профилът ни е бил одобрен от администратора;

index() - основно действие на контролера. Извиква необходимите модели за да покаже нужната информация за избран клас според зададен учебен период. Съдържа GET и POST заявки за извличане добавяне, промяна и изтриване на оценки по избрани предмети;

create() - действие, което извлича налични класове, учебни периоди и предмети и POST заявки за тяхното добавяне.

ProfileController.php -

index() - основно действие на контролера, което извлича от модела информацията за влезлия в системата потребител.

view(int \$id) – действие, което проверява дали имаме създаден профил и ако е така извежда пълна информация за потребителя.

create() - Съдържа POST заявки за попълване на профилната информация.

studentProfile(\$studentID) – Извлича данни за студент. Съдържа POST заявка за добавяне на профилна снимка. Снимките, които се добавят се съхраняват във файловата система.

teacherProfile(\$teacherID) - Извлича данни за учителя относно класовете и предметите по които преподава. Класен ръководител и има ли отговорник на класа. Съдържа POST заявки за добавяне и изтриване на предмет по които преподава; на кой клас води; добавяне на класен ръководител и отговорник на класа.

32Стр.

parentProfile() - Извлича данни за родителя. Чрез GET заявка по зададено ЕГН, намира ученик, които може да бъде добавен със съответната роднинска връзка.

edit(\$id) – действие което позволява редактирането на личната информация за всеки потребител.

delete(\$id) - изтрива профил по зададено ID.

4.1.3.Директория model

В тази директория са файловете, които се свързват, извличат и обработват данните от базата. В модела се намират данните и правилата, приложими към тези данни, които представляват различни концепции. Във всяка софтуерна система, всичко се моделира като данни, които ние обработваме по определен начин. – Примерно: Дали поисканото е потребителско име, съобщение или книга за приложението? – датата не може да бъде в бъдещето, е-пощата, трябва да има определен формат, името не може да бъде повече от х знака, и т.н. Моделът дава на контролера такива данни каквито е поискал потребителят (съобщение, списък с книги, фото албум и т.н.). Този модел на данните ще бъде един и същ, без значение как ние искаме да го представим на потребителя, това е защото ние можем да избираме от всеки наличен изглед(view), за да го представим. Моделът съдържа най-важната част от логиката на нашето приложение, логиката, която се отнася до проблема, с който ние се занимаваме (форум, магазин, банка и т.н.). Контролерът съдържа повече вътрешно-организационна логика за самото приложение. За обслужване на разработеното приложение имаме определени следните няколко модела:

BaseModel.php - осъществява връзката с базата данни;

ClassModel.php – съдържа функции за извличане, добавяне, промяна, изтриване на информация за всяка ученическа група. Всичко, което се отнася до учениците, учителите и тяхната дейност, преминава през този модел.

EventsModel.php – отговаря за статиите, които учителите публикуват, изтриват или

променят.

HomeModel.php – извежда информацията достъпна до всички потребители на приложението;

NotificationsModel.php – моделът, който отговаря за определянето на статута на потребителите. Използва се от администратора за да определи правата на достъп за всеки от тях.

ProfileModel.php - съдържа функции за извличане, добавяне, промяна, изтриване на информация за всеки профил от различните категории потребители.

33Стр.

UsersModel.php – грижи се за оторизацията на потребителите при тяхното вписване в системата. Извежда пълната информация, необходима на системата, за нейното правилно функциониране.

4.1.4.Директория view

Изгледът предлага различни начини за представяне на получените от модела данни. Те могат да бъдат шаблони, където се попълват получените данни. Може да има няколко различни изгледа и контролера да трябва да реши кой да използва. В тази директория се съдържат изгледите на контролерите. Всеки контролер има по няколко изгледа. Така например контролер *profile*, има изгледи за създаване на профил, за изтриване, редакция. Също така този контролер има изгледи за генериране съдържанието на всеки отделен тип потребител.

4.2.Описание на програмен код.

4.2.1.Регистрация /вход/изход.

Регистрация

View/users/register.php

...

```
<form role="form" method="post">
    <div class="form-group">
        <label for="username">Потребителско име:</label>
        <input id="username" type="text" name="username" autofocus
class="form-control">
        <p class="help-block">Изписва се с малка буква без разстояние.</p>
        <label for="password">Парола:</label>
        <input id="password" type="password" name="password" class="form-
control">
        <p class="help-block">Цифри и букви.</p>
        <label for="email">email:</label>
        <input id="email" type="text" name="email" class="form-control">
        <p class="help-block">Цифри и букви. всичко, както и @ </p>

        <button type="submit" value="registration" class="btn btn-
default">Регистрация</button>
```

```

        <a href="<?=APP_ROOT?>/users/login">[Към входа]</a>
    </div>
</form>

...

```

Дефинирана е форма, която позволява изращането на данни към web сървъра. В случая имаме данни касаещи регистрацията на потребител в системата. Методът на изпращане е POST, при който данните към сървъра се изпращат самостоятелно. Елементът <input> позволява въвеждането на информация от страна на потребителя. Атрибута „type“ задава

34Стр.

типа на елемента, който се въвежда, а „name“ задава името на елемента, като име на променливата към сървъра. Бутоните се асоциират с определени действия при кликане. <button type="submit">...</button> изпраща въведените във входните полета данни към сървъра. Данните, които се изпращат се обработват от контролера.

Controllers/UsersController.php

```

...
public function register()
{
    if($this->isPost){
        $username = $_POST['username'];
        if(strlen($username) < 3 || strlen($username) > 25){
            $this->setValidationError("username",
                                   "неправилно потребителско име");
        }
        $password = $_POST['password'];
        if(strlen($password) < 3 || strlen($password) > 25){
            $this->setValidationError("password", "неправилна парола");
        }
        $email = $_POST['email'];
        if(strlen($email) < 3 || strlen($email) > 100){
            $this->setValidationError("email", "грешна ел.поща");
        }
        if($this->formValid()){
            if($this->model->register($username, $password, $email)){
                $this->addInfoMessage("Регистрацията е успешна, но се налага да
изчакате одобрение от Администратора.");
                $this->redirect("");
            }
            else{
                $this->addErrorMessage("Имаме проблем с Вашата регистрация.");
            }
        }
    }
}
...

```

Ако имаме POST заявка функцията ще обработи входните данни. На променливата \$username присвояваме стойността от получената POST заявка с атрибут username. След това

извършваме задължителната валидация на получените данни. Тази проверка е важна, тъй като предотвратява от възможността умишлено или не, някои да съботи системата. Данните, които получаваме от формата трябва да бъдат точно такива каквито очакваме. В случая този критерий е сведен до определяне на дължината на получените данни. Проверяваме дали броя на символите е по-малък от 3 или по-голям от 25. Ако това условие е изпълнено нямаме `ValidationError` и продължаваме нататък. Ако моделът `register` върне `true`, значи че си е свършил работата. Извеждаме съобщение за успешна регистрация и препращаме потребителя към началната страница. Ако моделът `register` върне `false`,

35Стр.

извеждаме съобщение че не има никакъв проблем с регистрацията.

Models/UsersModel.php

...

```
public function register(string $username, string $password, string $email)
{
    $password_hash = password_hash($password, PASSWORD_DEFAULT);
    $statement = self::$db->prepare(
        "INSERT INTO users(username, password_hash, email) VALUES (?, ?, ?)";
    $statement->bind_param("sss", $username, $password_hash, $email);
    $statement->execute();
    if($statement->affected_rows !=1)
        return false;
    $userid = self::$db->query("SELECT LAST_INSERT_ID()")->fetch_row()[0];
    return $userid;
}
```

...

В модела използваме функцията `password_hash` за да криптираме паролата преди да я запишем в базата данни. Обръщаме се към базата с `self::$db`, и добавяме получените записи, чрез функцията `bind_param`.

Вход/изход

View/users/login.php

...

```
<form role="form" method="post">
    <div class="form-group">
        <label for="username">Потребителско име:</label>
        <input id="username" type="text" name="username" class="form-control"
autofocus>
```

```

<p class="help-block">Изписва се с малка буква без разстояние.</p>
<label for="password">Парола:</label>
<input id="password" type="password" name="password" class="form-
control">
<p class="help-block">Цифри и букви.</p>
<button type="submit" value="Login" class="btn btn-
default">Влез</button>
<a href="<?=APP_ROOT?>/users/register">[Нямате регистрация?]</a> </div>
</div>
</form>...

```

Стандартна форма за попълване на потребителско име и парола, която се изпраща до UsersController и действие login.

36Стр.

Controllers/UsersController.php

```

public function login()
{
    if($this->isPost) {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $loggedUserId = $this->model->login($username, $password);
        if($loggedUserId) {
            $_SESSION['username'] = $username;
            $_SESSION['user_id'] = $loggedUserId;
            $checkStatPosition = $this->model->getPositionById($loggedUserId);
            $_SESSION['positionID'] = $checkStatPosition['positionID'];
            $_SESSION['positiontype'] = $checkStatPosition['type'];
            $_SESSION['ustatus'] = $checkStatPosition['status'];

            $profileBy = $this->model->getProfileById($loggedUserId);
            $profileID = $profileBy['pid'];
            $_SESSION['profileID'] = $profileBy['pid'];
            $_SESSION['pstatus'] = $profileBy['pstatus'];
            if (!$checkStatPosition['positionID']) {
                $this->addErrorMessage("Моля изчакайте одобрение от администратора");
                $this->redirectUrl("../profile/create");
            }
        } else {
            if($checkStatPosition['positionID'] == 1) {
                $this->addInfoMessage("Добре дошъл Администратор " .
$profileBy['Fname']. $profileBy['Lname'].", разполагай се! :)");
                $this->redirect("");
            }
            if($checkStatPosition['positionID'] == 2) {
                $personsId = $profileBy['pid'];
                $classGrpStp = $this->model->getClsGrpStpBy($personsId);
                $_SESSION['classgrp'] = $classGrpStp['classgroupID'];
                $_SESSION['classstp'] = $classGrpStp['studyperiodID'];
                $position = "students";
                $studentID = $this->model->setPosition($profileID,$position);
                $_SESSION['studentID'] = $studentID;

                $this->addInfoMessage("Здравей " . $profileBy['Fname'].", ученик в
: :)");
                $this->redirectUrl("../profile/studentProfile/" . $profileBy['pid']);
            }
            if($checkStatPosition['positionID'] == 3) {
                $position = "teachers";
                $teacherID = $this->model->setPosition($profileID,$position);
                $_SESSION['teacherID'] = $teacherID;
                $this->addInfoMessage("Добър деннини учителю :)");
                $this->redirectUrl("../profile/"); //raboti
            }
        }
    }
}

```

```

    }
    if($cekStatPosition['positionID'] == 4) {
        $position = "parents";
        $parentID = $this->model->setPosition($profileID,$position);
        $_SESSION['parentID'] = $parentID;
        $this->addInfoMessage("Здравейте родители");
        $this->redirect("");
    }
}
}
}...

```

При наличиена получена POST заявка, на две променливи присвояваме получените стойности и ги изпращаме към Login модела. Ако върнатата от модела стойност е истина, значи че

37Стр.

въведените от потребителя потребителско име и парола са регистрирани в системата. В променливата \$_SESSION записваме някой от данните за влезналия потребител. Тези данни помагат за „предвиждането“ на потребителя през приложението. Прави се проверка за това дали потребителят е попълнил профилната си информация. По-надолу в кода проверяваме дали администраторът е определил роля за съответния потребител и ако е така го препращаме към съответния потребителски модул.

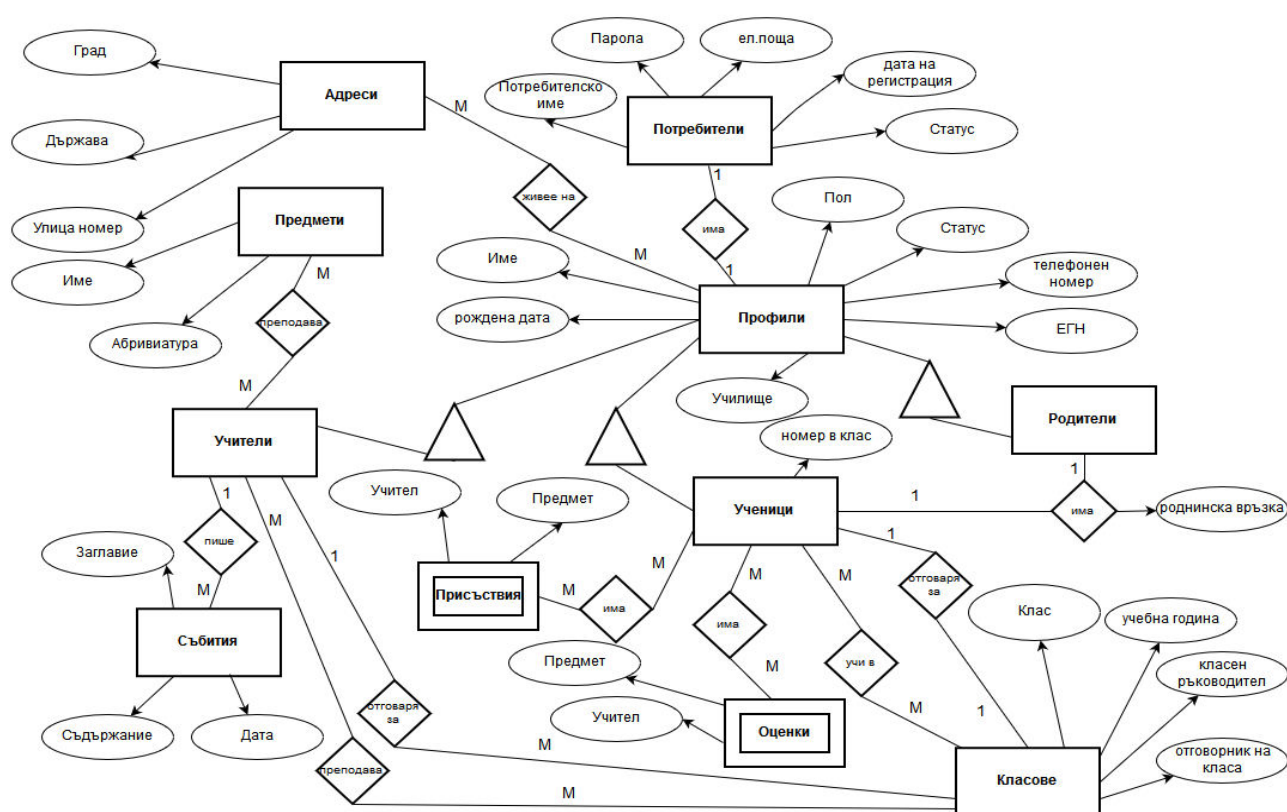
5.Изводи

Както с всяка софтуерна система, така и в разглежданата система съществува множество от функционалности, които все още не са създадени или въведени в експлоатация. Ако системата се развива добре, списъкът от допълнителни функционалности няма да намалее въпреки работата. Най-важна е оценката на потребителя, което ще определи и бъдещото развитие на системата. Тази система няма претенциите да замени напълно ученическите картони и бележници. По-скоро претендира за удобство и гъвкавост, която може да предостави на своите потребители.

6.Исползвана литература.

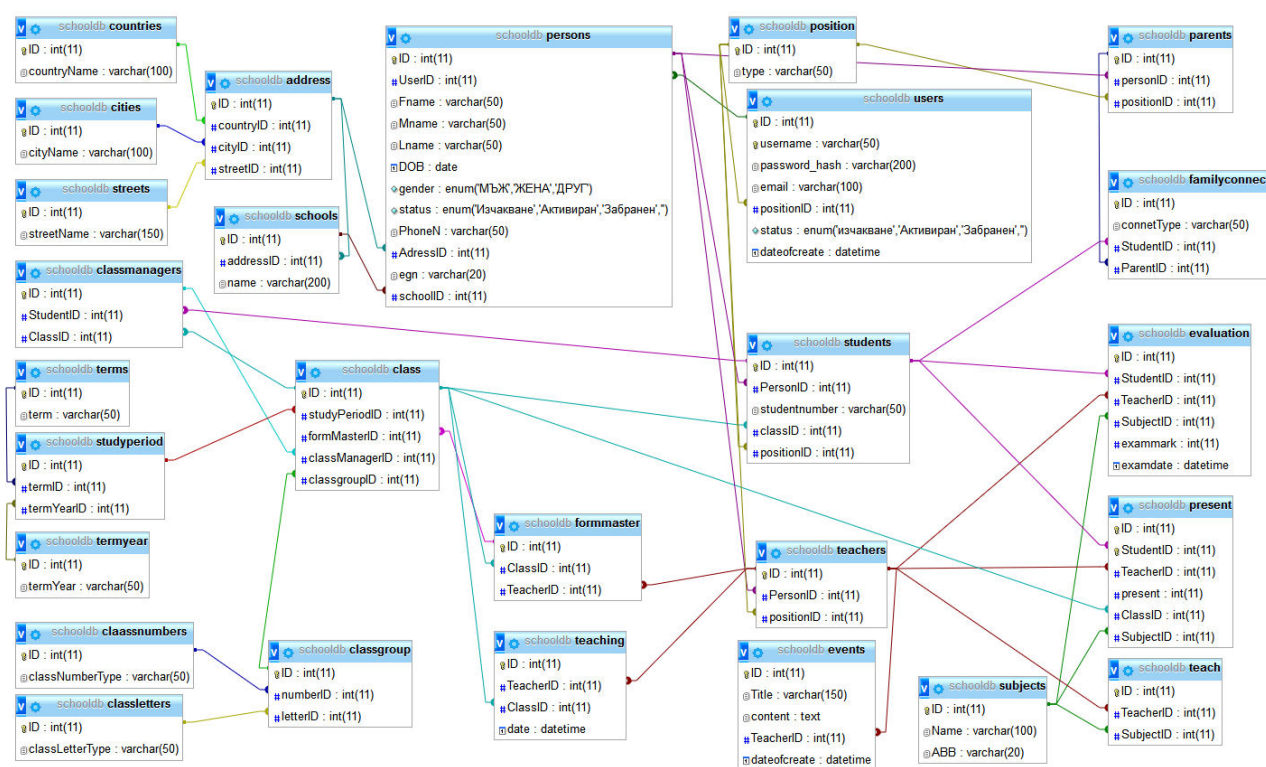
- 1.PHP7 & MySQL практическо програмиране – Денис Колисниченко – 2016г.
- 2.www.php.net -
- 3.<http://www.w3schools.com/>
- 4.<https://softuni.bg/>

7.Приложения



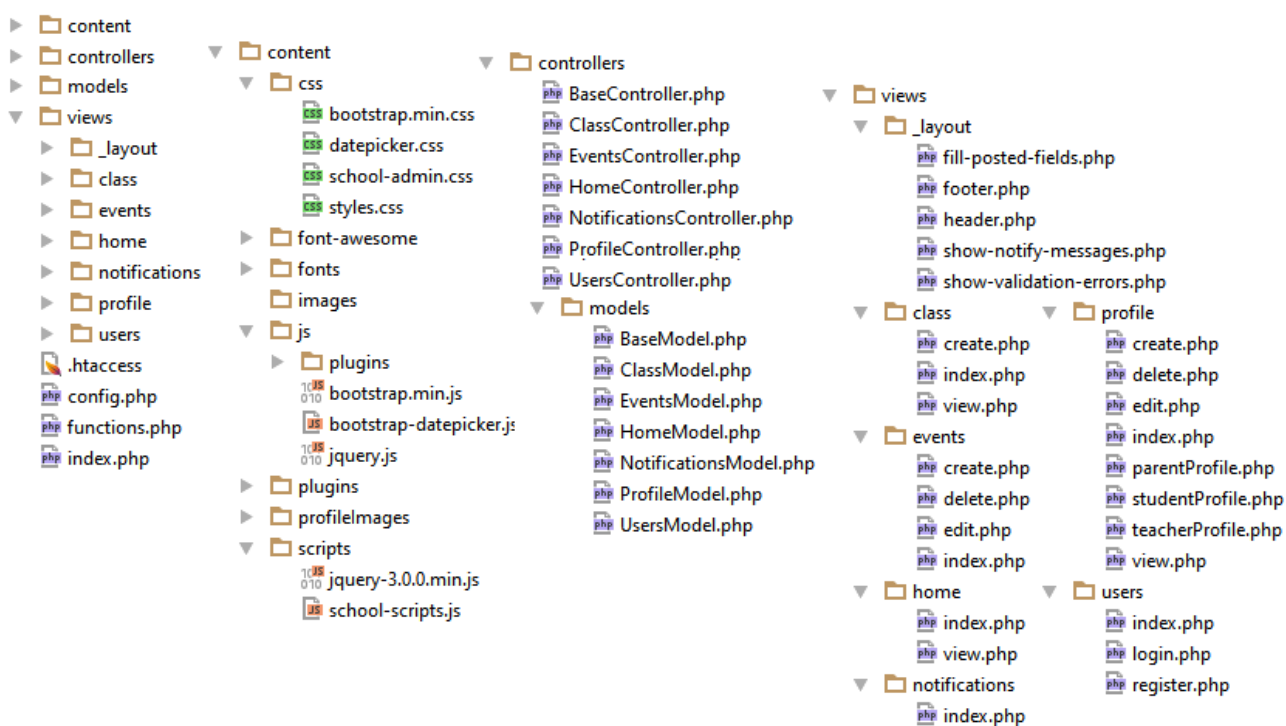
Приложение 1 . Концептуален модел

Приложение 2



. Реляционна схема на базата данни

Приложение 3



Структура проекта

Програмен код:

ClassController.php

```

<?php
class ClassController extends EventsController
{
    function onInit()
    {
        $this->authorize();
        if ($_SESSION['ustatus'] == 'Активиран' && $_SESSION['pstatus'] == 'Активиран' ){
        }
        else{
            $this->addInfoMessage("Може би трябва да бъдете одобрен от Администратора");
            $this->redirect("");
        }
    }
    public function index()
    {
        $school = $this->model->getSchool();
        $this->school = $school;
        $allclass = $this->model->getClassGroup();
        $this->classAll = $allclass;
        $classgroup = $this->model->getClassGroup();
        $this->classgroup = $classgroup;
        $classNumber = $this->model->selectClassNumber();
        $this->classNumber = $classNumber;

        $classLetter = $this->model->selectClassLetter();
        $this->classLetter = $classLetter;
        $studyperiod = $this->model->getStudyPeriod();
        $this->studyperiod = $studyperiod;
        $subjects = $this->model->getSubjects();
        $this->subjects = $subjects;

        if (isset($_GET['availablePeriod'])){
            $period = $_GET['availablePeriod'];
            $classByNumber = $this->model->selectClassByNumber($period);
            $this->classByNumber = $classByNumber;
            $subjects = $this->model->getSubjects();
            $this->subjects = $subjects;
        }
        if (isset($_GET['classletter']) && isset($_GET['period']) ){
            $TeacherID = $_SESSION['teacherID'];
            $classID = $_GET['classletter'];
            $classgrp = $_GET['classletter'];
            $period = $_GET['period'];

            $classByID = $this->model->getClassBy($classID, $period);
            $this->classbyid = $classByID;
            $subjects = $this->model->getSubjects();
            $this->subjects = $subjects;
            $teachercon = $this->model->teachersConnectsBy($classgrp, $period, $TeacherID);
            $this->teachercon = $teachercon;
            $students = $this->model->getStudentBy($classgrp, $period);
            $this->students = $students;
            if ($this->isPost){
                if (isset($_POST['mark']) && isset($_POST['TeacherID']) && isset($_POST['StudentID']) &&
                isset($_POST['SubjectID'])){
                    $mark = $_POST['mark'];
                    $teacherID = $_POST['TeacherID'];
                    $studentID = $_POST['StudentID'];
                    $subjectID = $_POST['SubjectID'];
                    if($this->model->setExamMark($studentID, $teacherID, $subjectID, $mark)){
                        if ($this->formValid()){
                            $this->addInfoMessage("Студентът беше оценен успешно.");
                            $this->redirectToUrl("/school/class/index/?classletter=".$classID."&period=".$period);
                        }
                    }
                }
            }
        }
    }
}

```

```

        $this->addErrorMessage("Грешка.");
    }
}

if (isset($_POST['editmark']) && isset($_POST['MarkID'])) {
    $markID = $_POST['MarkID'];
    $editmark = $_POST['editmark'];
    if($this->model->updateEvaluation($editmark, $markID)) {
        if ($this->formValid()) {
            $this->addInfoMessage("Променихте оценка");
            $this->redirectUrl("/school/class/index/?classletter=".$_classID."&period=".$_period);
        }
        else {
            $this->addErrorMessage("Грешка.");
        }
    }
}

if (isset($_POST['MarkIDdelete'])) {
    $markIDdelete = $_POST['MarkIDdelete'];

    if($this->model->deleteEvaluation($markIDdelete)) {
        if ($this->formValid()) {
            $this->addInfoMessage("Променихте оценка");
            $this->redirectUrl("/school/class/index/?classletter=".$_classID."&period=".$_period);
        }
        else {
            $this->addErrorMessage("Грешка.");
        }
    }
}

}

if ((isset($_GET['student']) && isset($_GET['subject']))) {
    $studentid = $_GET['student'];

    $studentbyid = $this->model->getStudentByID($studentid);
    $this->studentbyid = $studentbyid;
    $this->HomeClass = $this->model->getHomeClassBy($studentid);
    $subject = $_GET['subject'];
    $subjects = $this->model->getSubjectBy($subject);
    $this->subjects = $subjects;
    $studentMarks = $this->model->getStudEvaluationBy($studentid,$subject);
    $this->studentmarks = $studentMarks;
    $this->teacherBySubject = $this->model->teachersteachingBy($subject);
    if (isset($_GET['evaluation'])) {
        $markID = $_GET['evaluation'];
        $this->examMarkID = $this->model->getEvaluationBy($markID);
    }
    if ($this->isPost) {
        if (isset($_POST['student']) && isset($_POST['subject']) && isset($_POST['evaluation']) &&
isset($_POST['teacher'])) {
            $studentpost = $_POST['student'];// validaciq zadyljitelno, ifove mifove kvoto e neobhodimo
            $subjectpost = $_POST['subject'];//syshto
            $mark = $_POST['evaluation'];
            $teacherid = $_POST['teacher'];
            if($this->model->setExamMark($studentpost, $teacherid, $subjectpost, $mark)){
                if ($this->formValid()) {
                    $this->addInfoMessage("Студентът беше оценен успешно.");
                    $this->redirectUrl("/school/class/index/?student=".$_studentid."&subject=".$_subject);
                }
                else {
                    $this->addErrorMessage("Грешка.");
                }
            }
        }
    }

    if (isset($_POST['editmark']) && isset($_POST['MarkID'])) {
        $markID = $_POST['MarkID'];
        $editmark = $_POST['editmark'];
        if($this->model->updateEvaluation($editmark, $markID)) {
            if ($this->formValid()) {
                $this->addInfoMessage("Променихте оценка");
                $this->redirectUrl("/school/class/index/?student=".$_studentid."&subject=".$_subject);
            }
            else {
                $this->addErrorMessage("Грешка.");
            }
        }
    }
}

if (isset($_POST['MarkIDdelete'])) {
    $markIDdelete = $_POST['MarkIDdelete'];
    if($this->model->deleteEvaluation($markIDdelete)) {
        if ($this->formValid()) {
            $this->addInfoMessage("Изтрихте оценка");
            $this->redirectUrl("/school/class/index/?student=".$_studentid."&subject=".$_subject);
        }
        else {
            $this->addErrorMessage("Грешка.");
        }
    }
}
}
}

```



```

        if ($this->formValid()){
            $userId = $_SESSION['user_id'];
            if($this->model->create($title, $content, $userId)){
                $this->addInfoMessage("Event created.");
                $this->redirect("events");
            } else {
                // $this->addErrorMessage("Error:cannot create event.");
            }
        }
    }

    public function edit(int $id)
    {
        if($this->isPost){
            $title = $_POST['event_title'];
            if(strlen($title) < 1){
                $this->setValidationError("event_title", "Title cannot be empty!");
            }
            $content = $_POST['event_content'];
            if(strlen($content) < 1){
                $this->setValidationError("event_content", "Conent cannot be empty!");
            }
            $date = $_POST['event_date'];
            $dateRegex = '/^\d{2,4}-\d{1,2}-\d{1,2}( \d{1,2}:\d{1,2}:\d{1,2})?$/';
            if (!preg_match($dateRegex, $date)){
                $this->setValidationError("event_date", "Invalid date!");
            }
            $userid = $_POST['user_id'];
            if ($userid <= 0 || $userid > 10000){
                $this->setValidationError("user_id", "Invalud autohr user ID!");
            }
            if ($this->formValid()) {
                if ($this->model->edit($id, $title, $content, $date, $userid)) {
                    $this->addInfoMessage("Post edited.");
                } else {
                    $this->addErrorMessage("Error: cannot edit post.");
                }
                $this->redirect('events');
            }
        }
        $event = $this->model->getById($id);
        if (!$event) {
            $this->addErrorMessage("Error: post does not exists.");
            $this->redirect("events");
        }
        $this->event = $event;
    }

    public function delete(int $id)
    {
        if($this->isPost){
            if($this->model->delete($id)){
                $this->addInfoMessage("Event deleted.");
            } else{
                $this->addErrorMessage("Error: cannot delete event.");
            }
            $this->redirect('events');
        } else {
            $event = $this->model->getById($id);
            if(!$event){
                $this->addInfoMessage("Error: event does not exist.");
                $this->redirect("events");
            }
            $this->event = $event;
        }
    }
}
?>

```

HomeController.php

```

<?php
class HomeController extends BaseController
{
    function index() {

        // $this->school = $this->model->getSchool();

        $events = $this->model->getLatestEvents(6);
        $this->events = array_slice($events,0,4 );
        $this->eventsSidebar = $events;
        $school = $this->model->getSchool();
        $this->school = $school;
    }

    function view(int $id) {
        $this->school = $this->model->getSchool();
        $event = $this->model->getEventById($id);
    }
}

```

```

        if(!$event){
            $this->addErrorMessage("Error: invalid event id.");
            $this->redirect("");
        }
        $this->event = $event;
    }
}

```

NotificationsController.php

```

<?php
class NotificationsController extends BaseController
{
    function onInit()
    {
        $this->authorize();
        if ($_SESSION['ustatus'] == 'Активиран' && $_SESSION['pstatus'] == 'Активиран' ){
        }
        else{
            $this->addInfoMessage("Може би трябва да бъдете одобрен от Администратора");
            $this->redirect("");
        }
    }
    /**
     * @return boolean
     */
    public function index()
    {
        $school = $this->model->getSchool();
        $this->school = $school;
        $userST = $this->model->getAllUsers();
        $this->userST = $userST;

        $records = 0;
        $N = 6;
        $pro = $this->model->chekPro();
        $this->pro = $pro; //Общ брой записа след заявката
        $_SESSION['asd'] = $records + $N;
        $_SESSION['ass'] = $pro;
        if (isset($_GET['page'])) {
            if (!isset($_GET['page'])) {
                $page = 0;
            } else
                $page = $_GET['page'];
            $records = $page * $N;
            $profiles = $this->model->getAllProfile($records, $N);
            $this->profiles = $profiles;

            if (isset($_GET['search'])) {
                $search = $_GET['search'];
                $search = $this->model->searchEGN($search);
                $this->search = $search;
            }

            if ($this->isPost) {
                $ustatus = $_POST['ustatus'];
                $userid = $_POST['useridActive'];
                $pstatus = $_POST['pstatus'];
                $posid = $_POST['posID'];
                $profileID = $_POST['profileID'];

                if ($this->formValid()) {
                    if (isset($ustatus))
                        $this->model->addUstatus($ustatus, $userid);
                    if (isset($pstatus))
                        $this->model->addPstatus($pstatus, $userid);
                    if (isset($posid))
                        $this->model->setPosition($posid, $userid);
                    if (isset($profileID)) {
                        if ($posid == 1) {
                            $this->redirect("notifications");
                        }
                        if ($posid == 2) {
                            if ($this->model->insertStudent($posid, $profileID)){
                                $this->redirect("notifications");
                                $this->addInfoMessage("Записът беше променен.");
                            }
                        }
                        if ($posid == 3) {
                            if ($this->model->insertTeacher($posid, $profileID)){
                                $this->redirect("notifications");
                                $this->addInfoMessage("Записът беше променен.");
                            }
                        }
                        if ($posid == 4) {
                            if ($this->model->insertParent($posid, $profileID)){

```



```

        $this->addInfoMessage("Снимката не може да бъде качена");
        // if everything is ok, try to upload file
    } else {
        if (move_uploaded_file($image_tmp_name, $target_file)) {
            $this->redirect("profile");
            $this->addInfoMessage("Профилната снимка успешно качена");
        } else {
            $this->addInfoMessage("Снимката не може да бъде качена");
        }
    }
}

function view(int $id)
{
    $profileBy = $this->model->getProfileById($id);
    if(!$profileBy){
        $this->addErrorMessage("Не сте регистриран напълно в системата");
        $this->redirect("");
    }
    $profile = $this->model->getProfileById($id);
    if(!$profile){
        $this->addErrorMessage("невалиден профил");
        $this->redirect("");
    }
    $this->profile = $profile;
    $school = $this->model->getSchool();
    $this->school = $school;
    $profilePosition = $this->model->getPositiondByUser($id);
    $this->profilePosition = $profilePosition;
    $personprofile = $this->model->getPersonByUserId($id);
    $this->personprofile = $personprofile;
    $personcls = $this->model->getPersonClass($id);
    $this->personcls = $personcls;
}

public function upload()
{
    $uploads_dir = 'school/content/upload';
    $tmp_name = $_FILES["fileToUpload"]["tmp_name"];
    $name = basename($_FILES["fileToUpload"]["name"]);
    move_uploaded_file($tmp_name, "$uploads_dir/$name");
}

public function create()
{
    $userID = $_SESSION['user_id'];
    $profileBy = $this->model->getProfileById($userID);
    $this->profileby = $profileBy;
    $school = $this->model->getSchool();
    $this->school = $school;
    if ($this->isPost) {
        $fname = $_POST['person_fname'];
        $mname = $_POST['person_mname'];
        $lname = $_POST['person_lname'];
        $dateofbirth = $_POST['person_date'];
        $egn = $_POST['egn'];
        $country = $_POST['country'];
        $city = $_POST['city'];
        $street = $_POST['street'];
        $gender = $_POST['gender'];
        $phoneN = $_POST['phonenumber'];

        if ($this->formValid()) {
            $this->model->createAddress($country, $city, $street);
            $addressID = $_SESSION['addressID'];
            if ($this->model->createProfile($userID, $fname, $mname, $lname, $dateofbirth, $egn,
$addressID, $gender, $phoneN)) {
                $this->addInfoMessage("Профилът беше добавен");
                $this->redirect("");
                // $this->redirectToUrl("../profile/teacherProfile/$userID");
            } else {
                // $this->addErrorMessage("Error:cannot create event.");
            }
        }
    }
}

public function studentProfile($studentID)
{
    $school = $this->model->getSchool();
    $this->school = $school;
    $userID = $_SESSION['user_id'];
    $profileBy = $this->model->getProfileById($userID);
    $this->profileby = $profileBy;
    $studyperiod = $this->model->getStudyPeriod();
    $this->studyperiod = $studyperiod;
    $classgroup = $this->model->getClassGroup();
    $this->classgroup = $classgroup;
    if ($this->isPost) {
        if (isset($_POST['upload'])) {
            $target_dir = "content/profileImages/";
            $image_name = $_FILES['image']['name'];

```

```

$image_size = $_FILES['image']['size'];
$image_tmp_name = $_FILES['image']['tmp_name'];
$target_file = $target_dir . $image_name;
$username = $_SESSION['username'];
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
$target_file = $target_dir . $username . "." . $imageFileType;
// Check if image file is a actual image or fake image
$check = getimagesize($_FILES["image"]["tmp_name"]);
if($check !== false) {
    $this->addInfoMessage("Снимката е във формат - ". $check["mime"]);
    $uploadOk = 1;
} else {
    $this->addInfoMessage("Файлът не е снимка.");
    $uploadOk = 0;
}
// Check if file already exists
if (file_exists($target_file)) {
    $this->addInfoMessage("Снимката вече съществува.");
    $uploadOk = 0;
}
// Check file size
if ($image_size > 500000) {
    $this->addInfoMessage("Снимката е твърде голяма.");
    $uploadOk = 0;
}
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
    && $imageFileType != "gif" ) {
    $this->addInfoMessage("Снимката не е в един от форматите JPG, JPEG, PNG или GIF ");
    $uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    $this->addInfoMessage("Снимката не може да бъде качена");
    // if everything is ok, try to upload file
} else {
    if (move_uploaded_file($image_tmp_name, $target_file)) {
        $this->redirect("profile");
        $this->addInfoMessage("Профилната снимка успешно качена");
    } else {
        $this->addInfoMessage("Снимката не може да бъде качена");
    }
}
}

if (isset($_POST['classgroup']) && isset($_POST['period']) && isset($_POST['stNumber'])) {
    $profileBy = $this->model->getProfileById($userID);
    $this->profileBy = $profileBy;
    $studentid = $profileBy['pid'];
    $classgr = $_POST['classgroup'];//validaciq
    $period = $_POST['period'];//validaciq
    $stnumber = $_POST['stNumber'];
    if ($this->formValid()) {
        if ($this->model->setClassStudents($classgr, $period, $stnumber, $studentid)) {
            $this->addInfoMessage("Класът беше добавен");
            $this->redirect("profile");
        } else {
            $this->addErrorMessage("Ня успешно създаване на студентски период.");
        }
    } else {
        // $this->addErrorMessage("Error:cannot create event.");
    }
}
}

}

public function teacherProfile($teacherID)
{
    $school = $this->model->getSchool();
    $this->school = $school;
    $userID = $_SESSION['user_id'];
    $profileBy = $this->model->getProfileById($userID);
    $this->profileby = $profileBy;
    if (isset($profileBy['FTeacher'])) {
        $getFormMaster = $this->model->getFormMaster($teacherID);
        $this->getFormMaster = $getFormMaster;
    }
    $subTeacher = $this->model->getSubjectsTeacher($teacherID);
    $this->subTeacher = $subTeacher;
    $subjects = $this->model->getSubjects();
    $this->subjects = $subjects;
    $classavailable = $this->model->getAvailableClass();
    $this->classavailable = $classavailable;
    $teachingClass = $this->model->getTeachingClass($teacherID);
    $this->teachingClass = $teachingClass;
    $teacherFormMaster = $this->model->getFormMaster($teacherID);
    $this->teacherFormMaster = $teacherFormMaster;
    $classWithoutFormMaster = $this->model->getClassWithoutFormMaster();
    $this->classWithoutFormMaster = $classWithoutFormMaster;
    $classManager = $this->model->getStudentTeachersClass($teacherID);
    $this->selectClassManager = $classManager;
    $chosenClassManager = $this->model->getClassManager($teacherID);
}

```

```

$this->chosenClassManager = $chosenClassManager;
if ($this->isPost){
    if (isset($_POST['subjectset'])){
        $subjectset = $_POST['subjectset'];
        if ($this->formValid()){
            if ($this->model->setTeach($teacherID,$subjectset)) {
                $this->addInfoMessage("Предмета, по който преподавател беше добавен");
                $this->redirectUrl("");
            }
            else {
                $this->addErrorMessage("Грешка, ");
            }
        }
    }
    if (isset($_POST['classInsert'])){
        $classInsertID = $_POST['classInsert'];
        if ($this->formValid()){
            if ($this->model->setTeaching($teacherID,$classInsertID)) {
                $this->addInfoMessage("Класът, на който преподавател беше добавен");
                $this->redirectUrl("");
            }
            else {
                $this->addErrorMessage("Грешка, ");
            }
        }
    }
    if (isset($_POST['formMasterInsert'])){
        $formMasterInsertID = $_POST['formMasterInsert'];
        if ($this->formValid()){
            if ($this->model->setFormMaster($teacherID,$formMasterInsertID)) {
                $this->addInfoMessage("Класът, на който преподавател беше добавен");
                $this->redirectUrl("");
            }
            else {
                $this->addErrorMessage("Грешка, ");
            }
        }
    }
    if (isset($_POST['classManagerInsert']) && isset($_POST['CMclassID'])){
        $classManagerInsertID = $_POST['classManagerInsert'];
        $classManagerID = $_POST['CMclassID'];
        if ($this->formValid()){
            if ($this->model->setClassManager($classManagerInsertID,$classManagerID)) {
                $this->addInfoMessage("Отговорникът на класа, който избрахте беше добавен");
                $this->redirectUrl("");
            }
            else {
                $this->addErrorMessage("Грешка, ");
            }
        }
    }
    if (isset($_POST['delSubject'])){
        $teacherID = $_SESSION['teacherID'];
        $deletesubject = $_POST['delSubject'];
        if ($this->model->deleteSubject($deletesubject)){
            if ($this->formValid()){
                $this->addInfoMessage("Предметът беше премахнат");
                $this->redirectUrl("/school/profile/teacherProfile/".$teacherID);
            }
            else {
                $this->addErrorMessage("Грешка.");
            }
        }
    }
    if (isset($_POST['delTeaching'])){
        $teacherID = $_SESSION['teacherID'];
        $deleteTeaching = $_POST['delTeaching'];
        if ($this->model->deleteTeaching($deleteTeaching)){
            if ($this->formValid()){
                $this->addInfoMessage("Периодът на преподаване беше премахнат");
                $this->redirectUrl("/school/profile/teacherProfile/".$teacherID);
            }
            else {
                $this->addErrorMessage("Грешка.");
            }
        }
    }
}

}

public function parentProfile()
{
    $school = $this->model->getSchool();
    $this->school = $school;
    $userID = $_SESSION['user_id'];
    $profileBy = $this->model->getProfileById($userID);
    $this->profileby = $profileBy;
    $id = $profileBy['parentID'];
    $kid = $this->model->kidProfile($id);
    if (!$kid){

```

```

        $this->addErrorMessage("нямате регистрирани роднини");
    }
    $this->kids = $kid;
    $this->Subjects = $this->model->getSubjects();
    if (!isset($_GET['subject']) || isset($_GET['subject'])) {
        if (isset($_GET['search'])) {
            $search = $_GET['search'];
            $sr = $this->model->searchEGN($search);
            if ($sr) {
                $this->search = $sr;
            }
            else {
                $this->addErrorMessage("няма намерени резултати");
                $this->redirectUrl("/school/profile/parentProfile/");
            }
        }
        if (isset($_GET['subject'])) {
            // $this->evaluation = $this->model->getStudEvaluationBy($personid, $subject);
        }
    }
    if (isset($_GET['search'])) {
        $search = $_GET['search']; //validaciq
        $search = $this->model->searchEGN($search);
        $this->search = $search;
    }
    if ($this->isPost) {
        if (isset($_POST['parentID']) && isset($_POST['studentID']) && isset($_POST['parentStudentType'])) {
            $parentID = $_POST['parentID'];
            $studentID = $_POST['studentID'];
            $parenttype = $_POST['parentStudentType'];
            if ($this->model->setParentConection($parenttype, $studentID, $parentID)) {
                if ($this->formValid()) {
                    $this->addInfoMessage("Добавихте успешно ученика");
                    $this->redirectUrl("/school/profile/parentProfile/".$parentID);
                }
                else {
                    $this->addErrorMessage("Грешка.");
                }
            }
        }
        if (isset($_POST['delFamilyConn'])) {
            $deletefamid = $_POST['delFamilyConn'];
            if ($this->model->deleteFamConn($deletefamid)) {
                $this->addInfoMessage("Премахнахте роднина.");
            }
            else {
                $this->addErrorMessage("Грешка");
            }
            $this->redirectUrl("/school/profile/parentProfile/");
        }
    }
}

public function edit($id)
{
    $school = $this->model->getSchool();
    $this->school = $school;

    $userID = $_SESSION['user_id'];
    $profileBy = $this->model->getProfileById($userID);
    $this->profileby = $profileBy;
    $profileID = $profileBy['pid'];
    $addressID = $profileBy['addressID'];
    if ($this->isPost) {
        $fname = $_POST['person_fname'];
        $mname = $_POST['person_mname'];
        $lname = $_POST['person_lname'];
        $dateofbirth = $_POST['person_date'];
        $segn = $_POST['egn'];
        $country = $_POST['country'];
        $city = $_POST['city'];
        $street = $_POST['street'];
        $gender = $_POST['gender'];
        $phoneN = $_POST['phonenumber'];
        if ($this->formValid()) {
            if ($this->model->editAddress($country, $city, $street, $addressID)) {
                $this->addInfoMessage("Местоположението беше сменено");
            }
            else {
                {
                    $this->redirect("profile");
                }
            }
            if ($this->model->editProfile($fname, $mname, $lname, $dateofbirth, $segn, $gender, $phoneN,
            $profileID)) {
                $this->addInfoMessage("Профилът беше променен");
                $this->redirect("");
                // $this->redirectUrl("../profile/teacherProfile/$userID");
            }
            else {
                // $this->addErrorMessage("Error:cannot create event.");
                $this->redirect("events");
            }
        }
    }
}

```

```

    }
}
public function delete(int $id)
{
    if($this->isPost){
        if($this->model->delete($id){
            $this->addInfoMessage("Профилът беше изтрит.");
        }
        else{
            $this->addErrorMessage("Грешка");
        }
        $this->redirect('notifications');
    }
    else {
        $profile = $this->model->getProfileById($id);
        if(!$profile){
            $this->addInfoMessage("Потребителят вече не съществува");
            $this->redirect("notifications");
        }
        $this->profile = $profile;
    }
}
}
?>

```

UsersController.php

```

<?php
class UsersController extends BaseController
{
    public function login()
    {
        if($this->isPost){

            $username = $_POST['username'];
            $password = $_POST['password'];
            $loggedUserId = $this->model->login($username, $password);
            if($loggedUserId){
                $_SESSION['username'] = $username;
                $_SESSION['user_id'] = $loggedUserId;
                $chkStatPosition = $this->model->getPositionById($loggedUserId);
                $_SESSION['positionID'] = $chkStatPosition['positionID'];
                $_SESSION['positiontype'] = $chkStatPosition['type'];
                $_SESSION['ustatus'] = $chkStatPosition['status'];

                $profileBy = $this->model->getProfileById($loggedUserId);
                $profileID = $profileBy['pid'];
                $_SESSION['profileID'] = $profileBy['pid'];
                $_SESSION['pstatus'] = $profileBy['pstatus'];
                if (!$chkStatPosition['positionID']) {
                    $this->addErrorMessage("Моля изчакайте одобрение от администратора");
                    $this->redirectToUrl("../profile/create");
                }
                else{
                    if($chkStatPosition['positionID'] == 1){
                        $this->addInfoMessage("Добре дошъл Администратор " .
$profileBy['Fname'].$profileBy['Lname'].", разполагай се! :)");
                        $this->redirect("");
                    }
                    if($chkStatPosition['positionID'] == 2){
                        $personsId = $profileBy['pid'];
                        $classGrpStp = $this->model->getClsGrpStpBy($personsId);
                        $_SESSION['classgrp'] = $classGrpStp['classgroupID'];
                        $_SESSION['classstp'] = $classGrpStp['studyperiodID'];
                        $position = "students";
                        $studentID = $this->model->setPosition($profileID,$position);
                        $_SESSION['studentID'] = $studentID;

                        $this->addInfoMessage("Здравей ".$profileBy['Fname'].", ученик в :");
                        $this->redirectToUrl("../profile/studentProfile/".$profileBy['pid']);
                    }
                    if($chkStatPosition['positionID'] == 3){
                        $position = "teachers";
                        $teacherID = $this->model->setPosition($profileID,$position);
                        $_SESSION['teacherID'] = $teacherID;
                        $this->addInfoMessage("Добър деннини учителю :)");
                        $this->redirectToUrl("../profile/"); //raboti
                    }
                    if($chkStatPosition['positionID'] == 4) {
                        $position = "parents";
                        $parentID = $this->model->setPosition($profileID,$position);
                        $_SESSION['parentID'] = $parentID;
                        $this->addInfoMessage("Здравейте родители");
                        $this->redirect("");
                    }
                }
            }
        }
    }
}

```

```

    }
}
public function register()
{
    if($this->isPost){
        $username = $_POST['username'];
        if(strlen($username) < 3 || strlen($username) > 25){
            $this->setValidationError("username", "invalid username");
        }
        $password = $_POST['password'];
        if(strlen($password) < 3 || strlen($password) > 25){
            $this->setValidationError("password", "invalid password");
        }
        $email = $_POST['email'];
        if(strlen($email) < 3 || strlen($email) > 100){
            $this->setValidationError("email", "invalid email");
        }
        if($this->formValid()){
            $userId = $this->model->register($username, $password, $email);
            if($userId){
                $this->addInfoMessage("Регистрацията е успешна, но се налага да изчакате одобрение от
Администратора.");
                $this->redirect(""); // todo: redirect to registration form
            }
            else{
                $this->addErrorMessage("Имаме проблем с Вашата регистрация.");
            }
        }
    }
}

public function logout()
{
    $this->addInfoMessage("Излязохте успешно от системата");
    $this->addErrorMessage("Моля изчакайте одобрение от администратора");
    session destroy();
    $this->redirect("");
}

public function index()
{
    $this->authorize();
    //$this->users = $this->model->getAll();
    if ($ _SESSION['ustatus'] == 'Активиран' && $_SESSION['pstatus'] == 'Активиран'){
        $this->users = $this->model->getAll();
        $school = $this->model->getSchool();
        $this->school = $school;
        $userId = $_SESSION['user_id'];
        $getPosition = $this->model->getPosition();
        $this->getPosition = $getPosition;
        if (isset($_GET['position'])){
            if ($_GET['position'] != $_SESSION['positionID'] ) {
                $this->addErrorMessage("нямате достатъчно права");
                $this->redirectToUrl("../");
            }
            if($_GET['position'] == 2 ){
                $classgrp = $_SESSION['classgrp'];
                $classstp = $_SESSION['classstp'];
                $studentinfo = $this->model->getAllStudents($classgrp,$classstp);
                $this->studentsinfo = $studentinfo;
                $subjects = $this->model->getSubjects();
                $this->subjects = $subjects;
            }
            if($_GET['position'] == 3 ){
                $this->teachers = $this->model->getAllTeachers();
            }
            if($_GET['position'] == 4 ){
                $this->parents = $this->model->getAllParents();
            }
            if($_GET['position'] >= 5 ){
                $this->addErrorMessage("няма такава страница");
                $this->redirectToUrl("../");
            }
        }
        if (isset($_GET['teachers'])){
            $this->teachers = $this->model->getAllTeachers();
        }
        if (isset($_GET['students'])){
            $this->students = $this->model->getAllStudents();
        }
        if (isset($_GET['parents'])){
            $this->parents = $this->model->getAllParents();
        }
    }
    else{
        $this->addInfoMessage("Може би трябва да бъдете одобрен от Администратора");
        $this->redirect("");
    }
}
}

```

BaseModel.php

```
<?php
abstract class BaseModel
{
    protected static $db;
    public function __construct()
    {
        if (self::$db == null) {
            self::$db = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
            self::$db->set_charset("utf8");
            if (self::$db->connect_errno) {
                die('Cannot connect to database');
            }
        }
    }
}
```

ClassModel.php

```
<?php
class ClassModel extends BaseModel
{
    public function getAll() : array
    {
    }
    public function getById(int $id)
    {
    }
    public function getSchool() : array
    {
        $statement = self::$db->prepare(
            "SELECT name, countryName, cityName, streetName " .
            "FROM schools INNER JOIN address ON schools.addressID = address.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "WHERE schools.ID = 1");
        //$statement->bind_param("i", $school );
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function checkTerm($term)
    {
        $stmTermCheck = self::$db->prepare("SELECT ID, term FROM terms WHERE term = ?");
        $stmTermCheck->bind_param("s", $term);
        $stmTermCheck->execute();
        $result = $stmTermCheck->get_result()->fetch_assoc();
        var_dump($result);
        if($result['term'] != $term){
            $stmInsertTerm = self::$db->prepare("INSERT INTO terms (term) VALUES (?)");
            $stmInsertTerm->bind_param("s", $term);
            $stmInsertTerm->execute();
            return $insertIdTerm = self::$db->insert_id;
        }
        else{
            return $insertIdTerm = $result['ID'];
        }
    }
    public function checkTermYear($termyear)
    {
        $stmTermYearCheck = self::$db->prepare("SELECT ID, termYear FROM termyear WHERE termYear = ?");
        $stmTermYearCheck->bind_param("s", $termyear);
        $stmTermYearCheck->execute();
        $result = $stmTermYearCheck->get_result()->fetch_assoc();
        //var_dump($result);
        if($result['termYear'] != $termyear){
            $stmInsertTermYear = self::$db->prepare("INSERT INTO termyear (termYear) VALUES (?)");
            $stmInsertTermYear->bind_param("s", $termyear);
            $stmInsertTermYear->execute();
            return $insertIdTermYear = self::$db->insert_id;
        }
        else{
            return $insertIdTermYear = $result['ID'];
        }
    }
    public function createStudyPeriod($termID, $termYearID): bool
    {
        $term = $this->checkTerm($termID);
        $termyear = $this->checkTermYear($termYearID);
        $checkTermAndYear = self::$db->prepare("SELECT termID , termYearID FROM studyperiod " .
            "WHERE termID = ? AND termYearID = ?");
        $checkTermAndYear->bind_param("ii", $term, $termyear );
    }
}
```

```

    $checkTermAndYear->execute();
    $lastResult = $checkTermAndYear->get result()->fetch assoc();
    if (($lastResult['termID'] != $term) && ($lastResult['termYearID'] != $termYear)){
        $stmInsertStudyPeriod = self::$db->prepare("INSERT INTO studyperiod (termID, termYearID) " .
            "VALUES (?, ?)");
        $stmInsertStudyPeriod->bind_param("ii", $term, $termYear);
        $stmInsertStudyPeriod->execute();
        if($stmInsertStudyPeriod->affected_rows !=1){
            return false;
        }else{
            $studyPeriodID = self::$db->query("SELECT LAST_INSERT_ID()")->fetch_row()[0];
            $_SESSION['studyperiod'] = $studyPeriodID;
            return $stmInsertStudyPeriod->affected_rows == 1;
        }
    }
    }
    else{
        return false;
    }
}

public function getStudyPeriod() : array
{
    $statement = self::$db->query(
        "SELECT studyperiod.ID as studyPeriodID ,term, termYear " .
        "FROM studyperiod INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "ORDER BY termyear.termYear ASC ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function checkNumber($number)
{
    $stmNumberCheck = self::$db->prepare("SELECT ID, classNumberType " .
        "FROM claassnumbers WHERE classNumberType = ?");
    $stmNumberCheck->bind_param("s", $number);
    $stmNumberCheck->execute();
    $result = $stmNumberCheck->get result()->fetch_assoc();
    if($result['classNumberType'] != $number){
        $stmInsertNumber = self::$db->prepare("INSERT INTO claassnumbers (classNumberType) VALUES (?)");
        $stmInsertNumber->bind_param("s", $number);
        $stmInsertNumber->execute();
        return $insertIdNumber = self::$db->insert_id;
    }
    else{
        return $insertIdNumber = $result['ID'];
    }
}

public function checkLetter($letter)
{
    $stmLetterCheck = self::$db->prepare("SELECT ID, classLetterType " .
        "FROM classletters WHERE classLetterType = ?");
    $stmLetterCheck->bind_param("s", $letter);
    $stmLetterCheck->execute();
    $result = $stmLetterCheck->get result()->fetch_assoc();

    if($result['letter'] != $letter){
        $stmInsertLetter = self::$db->prepare("INSERT INTO classletters (classLetterType) VALUES (?)");
        $stmInsertLetter->bind_param("s", $letter);
        $stmInsertLetter->execute();
        return $insertIdLetter = self::$db->insert_id;
    }
    else{
        return $insertIdLetter = $result['ID'];
    }
}

public function createClassGroup($numberID, $letterID) : bool
{
    $number = $this->checkNumber($numberID);
    $letter = $this->checkLetter($letterID);
    $stmInsertClassGroup = self::$db->prepare("INSERT INTO classgroup (numberID, letterID) " .
        "VALUES (?, ?)");
    $stmInsertClassGroup->bind_param("ii", $number, $letter);
    $stmInsertClassGroup->execute();
    if($stmInsertClassGroup->affected_rows !=1){
        return false;
    }else{
        $classGroupID = self::$db->query("SELECT LAST_INSERT_ID()")->fetch_row()[0];
        $_SESSION['classGroupID'] = $classGroupID;
        return $stmInsertClassGroup->affected_rows == 1;
    }
}

public function selectClassLetter()
{
    $statement = self::$db->query("SELECT classgroup.ID as classID , classletters.ID as letterID, " .
        "classLetterType as letterType, numberID " .
        "FROM classletters INNER JOIN classgroup ON classletters.ID = classgroup.letterID " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID;");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

```



```

}
public function selectClassNumber()
{
    $statement = self::$db->query("SELECT * FROM claassnumbers");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function selectClassByNumber($period)
{
    $statement = self::$db->query("SELECT claassnumbers.ID as claassnumbersID, classNumberType, numberID, " .
        "classgroup.ID as classID , classletters.ID as letterID, classLetterType as letterType " .
        "FROM class INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "WHERE studyPeriodID = $period");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getClassGroup() : array
{
    $statement = self::$db->query("SELECT classgroup.ID as classID, numberID, letterID, " .
        "classNumberType as numberType, classLetterType as letterType " .
        "FROM classgroup " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getClassGroupBy(int $classID) : array
{
    $statement = self::$db->query("SELECT classgroup.ID as classID, numberID, letterID, " .
        "classNumberType as numberType, classLetterType as letterType " .
        "FROM classgroup " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "WHERE classgroup.ID = $classID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function setExamMark($studentID, $teacherID, $subjectID, $mark)
{
    $stmInsertExamMak = self::$db->prepare(
        "INSERT INTO evaluation(StudentID, TeacherID, SubjectID, exammark)VALUE (?, ?, ?, ?)");
    $stmInsertExamMak->bind_param("iiii", $studentID, $teacherID, $subjectID, $mark);
    $stmInsertExamMak->execute();
    if($stmInsertExamMak->affected_rows !=1){
        return false;
    }else{
        return $stmInsertExamMak->affected_rows == 1;
    }
}

public function getEvaluationBy(int $markID) :array
{
    $statement = self::$db->prepare(
        "SELECT * FROM evaluation WHERE ID = $markID;");
    $statement->execute();
    $result = $statement->get_result()->fetch_assoc();
    return $result;
}

public function getStudEvaluationBy(int $personid, int $subject) :array
{
    $statement = self::$db->prepare(
        "SELECT * " . "FROM evaluation WHERE StudentID = $personid AND SubjectID = $subject " .
        "ORDER BY examdate DESC");
    $statement->execute();
    $result = $statement->get_result()->fetch_all(MYSQLI_ASSOC);
    return $result;
}

public function getStudentBy(int $classgrp, int $period) :array
{
    $statement = self::$db->query("SELECT persons.ID as PersonID, Fname, Lname, class.ID as ClassID,
studyPeriodID, " .
        " classgroupID, students.ID as studentID , studentnumber, users.ID as userID, username " .
        "FROM students INNER JOIN persons ON students.PersonID = persons.ID " .
        "INNER JOIN users ON persons.UserID = users.ID " .
        "INNER JOIN class ON students.classID = class.ID " .
        "WHERE classgroupID = $classgrp and studyPeriodID = $period");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function teachersConects(int $teacherID) : array
{
    $statement = self::$db->query("SELECT * " .
        "FROM teaching INNER JOIN class ON teaching.ClassID = class.ID " .
        "INNER JOIN teachers ON teaching.TeacherID = teachers.ID " .
        "INNER JOIN persons ON teachers.PersonID = persons.ID " .
        "INNER JOIN teach ON teachers.ID = teach.TeacherID " .
        "INNER JOIN subjects ON teach.SubjectID = subjects.ID " .
        "WHERE teachers.ID = $teacherID ;");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

```

```

}
public function teachersteachingBy(int $subject) : array
{
    $statement = self::$db->prepare("SELECT * " .
        "FROM teaching INNER JOIN class ON teaching.ClassID = class.ID " .
        "INNER JOIN teachers ON teaching.TeacherID = teachers.ID " .
        "INNER JOIN persons ON teachers.PersonID = persons.ID " .
        "INNER JOIN teach ON teachers.ID = teach.TeacherID " .
        "INNER JOIN subjects ON teach.SubjectID = subjects.ID " .
        "WHERE SubjectID = $subject ");
    $statement->execute();
    $result = $statement->get_result()->fetch_assoc();
    return $result;
}

public function teachersConectsBy(int $classgrp, int $period,int $teacherID) : array
{
    $statement = self::$db->query("SELECT * " .
        "FROM subjects INNER JOIN teach ON subjects.ID = teach.SubjectID " .
        "INNER JOIN teachers ON teach.TeacherID = teachers.ID " .
        "INNER JOIN persons ON teachers.PersonID = persons.ID " .
        "INNER JOIN teaching ON teachers.ID = teaching.TeacherID " .
        "INNER JOIN class ON teaching.ClassID = class.ID " .
        "WHERE classgroupID = $classgrp AND studyPeriodID = $period AND teachers.ID = $teacherID ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getStudentByID(int $studentID) :array
{
    $statement = self::$db->query("SELECT persons.ID as PersonID, Fname, Lname, class.ID as ClassID,
studyPeriodID, " .
        " classgroupID, studentnumber, users.ID as userID, username " .
        "FROM students INNER JOIN persons ON students.PersonID = persons.ID " .
        "INNER JOIN users ON persons.UserID = users.ID " .
        "INNER JOIN class ON students.classID = class.ID " .
        "WHERE students.ID = $studentID ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getClassBy(int $classID, int $period) : array
{
    $statement = self::$db->query("SELECT class.ID as ClassID, classNumberType as numberType, classLetterType
as letterType, " .
        "term, termYear, s.Fname as sFname, s.Lname as sLname, subjects.Name as subjectName, subjects.ABB as
subjectABB , " .
        "students.ID as StudentID, studentnumber as studNumber , classgroupID, studyPeriodID, " .
        "evaluation.ID as MarkID,exammark as Mark, examdate as MarkDate, " .
        "teachers.ID as TeacherID, t.Fname as tFname, t.Lname as tLname, " .
        "present.ID as presentID , present.StudentID as presentStudentID, present.TeacherID as
presentTeacherID, " .
        "present.ClassID as presenClassID, present.SubjectID as presentSubjectID, present , subjects.ID as
subjectID " .
        "FROM class INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "INNER JOIN classnumbers ON classgroup.numberID = classnumbers.ID " .
        "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN students ON class.ID = students.classID " .
        "INNER JOIN persons s ON students.PersonID = s.ID " .
        "LEFT JOIN evaluation ON students.ID = evaluation.StudentID " .
        "LEFT JOIN teachers ON evaluation.TeacherID = teachers.ID " .
        "LEFT JOIN persons t ON teachers.PersonID = t.ID " .
        "LEFT JOIN subjects ON evaluation.SubjectID = subjects.ID " .
        "LEFT JOIN present ON students.ID = present.StudentID " .
        "WHERE classgroupID = $classID AND studyPeriodID = $period " .
        "ORDER BY sFname, subjectName ASC ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function updateEvaluation($editmark,$markID) : bool
{
    $statement = self::$db->prepare("UPDATE evaluation SET exammark = ?, examdate = CURRENT_TIMESTAMP " .
        "WHERE ID = ?");
    $statement->bind_param("ii", $editmark, $markID);
    $statement->execute();
    return $statement->affected_rows == 1;
}

public function deleteEvaluation($markIDdelete) : bool
{
    $statement = self::$db->prepare(
        "DELETE FROM evaluation WHERE ID = ?");
    $statement->bind_param("i", $markIDdelete);
    $statement->execute();
    return $statement->affected_rows == 1;
}

public function createSubject($subject, $abb)
{
    $stmSubjectCheck = self::$db->prepare("SELECT ID, Name, ABB FROM subjects WHERE Name = ?");
    $stmSubjectCheck->bind_param("s", $subject);
    $stmSubjectCheck->execute();
    $result = $stmSubjectCheck->get_result()->fetch_assoc();
    if($result['Name'] != $subject || $result['ABB'] != $abb){

```

```

        $stmInsertSubject = self::$db->prepare("INSERT INTO subjects (Name, ABB) VALUES (?, ?)");
        $stmInsertSubject->bind_param("ss", $subject, $abb);
        $stmInsertSubject->execute();
        return $insertIdSubject = self::$db->insert_id;
    }
    else{
        return $insertIdSubject = $result['ID'];
    }
}

public function getSubjects() : array
{
    $statement = self::$db->query("SELECT * FROM subjects");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getSubjectBy(int $subject) :array
{
    $statement = self::$db->prepare(
        "SELECT * " . "FROM subjects WHERE ID = $subject ";
    $statement->execute();
    $result = $statement->get_result()->fetch_assoc();
    return $result;
}

public function getHomeClassBy(int $personid) :array
{
    $statement = self::$db->prepare("SELECT classgroupID , studyPeriodID " .
        "FROM class INNER JOIN students ON class.ID = students.classID " .
        "INNER JOIN persons ON students.PersonID = persons.ID " .
        "WHERE students.ID = $personid");
    $statement->execute();
    $result = $statement->get_result()->fetch_assoc();
    return $result;
}

public function edit(string $id, string $title, string $content, string $date, int $TeacherID) : bool
{
}

public function delete(int $id) : bool
{
}
}

```

EventsModel.php

```

<?php
class EventsModel extends BaseModel
{
    public function getAll() : array
    {
        $statement = self::$db->query(
            "SELECT * FROM events ORDER BY dateofcreate DESC ";
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getById(int $id)
    {
        $statement = self::$db->prepare(
            "SELECT * FROM events WHERE id = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }

    public function create(string $title, string $content, int $userID) : bool
    {
        $statement = self::$db->prepare(
            "INSERT INTO events(Title, content, TeacherID) VALUES (?,?,?)");
        $statement->bind_param("ssi", $title, $content, $userID);
        $statement->execute();
        return $statement->affected_rows == 1;
    }

    public function edit(string $id, string $title, string $content, string $date, int $TeacherID) : bool
    {
        $statement = self::$db->prepare("UPDATE events SET Title = ?, content = ?, " .
            " dateofcreate = ?, TeacherID = ? WHERE ID = ?");
        $statement->bind_param("sssii", $title, $content, $date, $TeacherID, $id);
        $statement->execute();
        return $statement->affected_rows >= 0;
    }

    public function delete(int $id) : bool
    {
        $statement = self::$db->prepare(
            "DELETE FROM events WHERE ID = ?");
        $statement->bind_param("i", $id);
    }
}

```

```

        $statement->execute();
        return $statement->affected_rows == 1;
    }
}

```

HomeModel.php

```

<?php
class HomeModel extends BaseModel
{
    function getLatestEvents(int $maxCount)
    {
        $statement = self::$db->query(
            "SELECT events.ID, Title, content, dateofcreate, persons.Fname " .
            "FROM events INNER JOIN teachers ON events.TeacherID = teachers.ID " .
            "INNER JOIN persons ON teachers.PersonID = persons.ID " .
            "ORDER BY dateofcreate DESC LIMIT " . $maxCount);
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getSchool() : array
    {
        $statement = self::$db->prepare(
            "SELECT name, countryName, cityName, streetName " .
            "FROM schools INNER JOIN address ON schools.addressID = address.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "WHERE schools.ID = 1");
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }

    function getEventById(int $id)
    {
        $statement = self::$db->prepare(
            "SELECT events.ID, Title, content, dateofcreate, persons.Fname " .
            "FROM events INNER JOIN teachers ON events.TeacherID = teachers.ID " .
            "INNER JOIN persons ON teachers.PersonID = persons.ID " .
            "WHERE events.ID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
}

```

NotificationsModel.php

```

<?php
class NotificationsModel extends BaseModel
{
    public function getSchool() : array
    {
        $statement = self::$db->prepare(
            "SELECT name, countryName, cityName, streetName " .
            "FROM schools INNER JOIN address ON schools.addressID = address.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "WHERE schools.ID = 1");
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }

    public function getAllUsers() : array
    {
        $statement = self::$db->query("SELECT users.ID as ID, username, password_hash, email, positionID as
posID,type, " .
            "users.status as ustatus, persons.status as pstatus " .
            "FROM users LEFT JOIN position ON users.positionID = position.ID " .
            "LEFT JOIN persons ON users.ID = persons.UserID " .
            "ORDER BY pstatus, posID, dateofcreate DESC LIMIT 0,4 ;");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }
}

```

```

public function getAllProfile($records, $N) : array
{
    $statement = self::$db->query(
        "SELECT users.ID as userid,username,users.status as ustatus,persons.ID as
pid,Fname,Mname,Lname,persons.status as pstatus " .
        ",DOB ,egn,gender,email,dateofcreate, countryName, cityName, streetName, position.ID as
posID,position.type " .
        "FROM persons INNER JOIN users ON persons.UserID = users.ID " .
        "INNER JOIN address ON persons.AddressID = address.ID " .
        "INNER JOIN countries ON address.countryID = countries.ID " .
        "INNER JOIN cities ON address.cityID = cities.ID " .
        "INNER JOIN streets ON address.streetID = streets.ID " .
        "LEFT JOIN teachers ON persons.ID = teachers.PersonID " .
        "LEFT JOIN students ON persons.ID = students.PersonID " .
        "LEFT JOIN parents ON persons.ID = parents.personID " .
        "LEFT JOIN position ON users.positionID = position.ID " .
        "ORDER BY pstatus ASC LIMIT $records,$N");

    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function RecordsProfile( $records, $N)
{
    $statement = self::$db->query("SELECT * " .
        "FROM persons ORDER BY id ASC LIMIT $records,$N");

    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function chekPro()
{
    $statement = self::$db->query("SELECT count(*) AS records " .
        "FROM persons");
    $result = mysqli_fetch_row($statement);
    return $rec_count = $result['0'];
}

public function addUstatus(int $ustatus, int $userid) : bool
{
    $statement = self::$db->prepare("UPDATE users SET status = ? WHERE users.ID = ?");
    $statement->bind_param("ii", $ustatus, $userid);
    $statement->execute();
    return $statement->affected_rows == 1;
}

public function addPstatus(int $pstatus, int $userid) : bool
{
    $statement = self::$db->prepare("UPDATE persons SET status = ? WHERE UserID = ?");
    $statement->bind_param("ii", $pstatus, $userid);
    $statement->execute();
    return $statement->affected_rows == 1;
}

public function setPosition(int $posid,int $userid) : bool
{
    $statement = self::$db->prepare("UPDATE users SET positionID = ? WHERE users.ID = ?");
    $statement->bind_param("ii", $posid, $userid);
    $statement->execute();
    return $statement->affected_rows == 1;
}

public function searchEGN($search) : array
{
    $statement = self::$db->query(
        "SELECT username,users.status as ustatus,persons.ID as pid,Fname,Mname,Lname,persons.status as
pstatus, " .
        "DOB,egn, gender, email,dateofcreate, countryName,cityName, streetName, position.ID as posID, " .
        "position.type, users.ID as userid ,students.ID as studentID, parents.ID as parentID, " .
        "teachers.ID as teacherID " .
        "FROM persons INNER JOIN address ON persons.AddressID = address.ID " .
        "INNER JOIN users ON persons.UserID = users.ID " .
        "INNER JOIN countries ON address.countryID = countries.ID " .
        "INNER JOIN cities ON address.cityID = cities.ID " .
        "INNER JOIN streets ON address.streetID = streets.ID " .
        "LEFT JOIN teachers ON persons.ID = teachers.PersonID " .
        "LEFT JOIN students ON persons.ID = students.PersonID " .
        "LEFT JOIN parents ON persons.ID = parents.personID " .
        "LEFT JOIN position ON users.positionID = position.ID " .
        "WHERE egn LIKE '%$search%'");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function insertStudent($posid, $profileID)
{
    $stmInsertTeacher = self::$db->prepare("INSERT INTO students (PersonID, positionID) VALUES (?,?)");
    $stmInsertTeacher->bind_param("ii",$profileID, $posid);
    $stmInsertTeacher->execute();
    if($stmInsertTeacher->affected_rows !=1){
        return false;
    }else{
        return $stmInsertTeacher->affected_rows == 1;
    }
}

```

```

    }
    public function insertTeacher($posid, $profileID)
    {
        $stmInsertTeacher = self::$db->prepare("INSERT INTO teachers ( PersonID, positionID) VALUES (?,?)");
        $stmInsertTeacher->bind_param("ii", $profileID,$posid);
        $stmInsertTeacher->execute();
        if($stmInsertTeacher->affected_rows !=1){
            return false;
        }else{
            return $stmInsertTeacher->affected_rows == 1;
        }
    }
    public function insertParent($posid, $profileID)
    {
        $stmInsertTeacher = self::$db->prepare("INSERT INTO parents ( personID, positionID) VALUES (?,?)");
        $stmInsertTeacher->bind_param("ii", $profileID,$posid);
        $stmInsertTeacher->execute();
        if($stmInsertTeacher->affected_rows !=1){
            return false;
        }else{
            return $stmInsertTeacher->affected_rows == 1;
        }
    }
}

```

ProfileModel.php

```

<?php
class ProfileModel extends BaseModel
{
    public function getAllAddress() : array
    {
        $statement = self::$db->query(
            "SELECT address.ID, countryName, cityName, streetName " .
            "FROM address INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "ORDER BY ID");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getSchool() : array
    {
        $statement = self::$db->prepare(
            "SELECT name, countryName, cityName, streetName " .
            "FROM schools INNER JOIN address ON schools.addressID = address.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "WHERE schools.ID = 1");
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }

    public function getAllCity() : array
    {
        $statement = self::$db->query(
            "SELECT * FROM cities ORDER BY cityName ASC ");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getAllProfile() : array
    {
        $statement = self::$db->query(
            "SELECT * " .
            "FROM persons INNER JOIN address ON persons.AddressID = address.ID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "INNER JOIN position ON position.ID = users.positionID");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function checkProfileSet() : array
    {
        $statement = self::$db->query("SELECT * " .
            "FROM persons INNER JOIN address ON persons.AddressID = address.ID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "INNER JOIN position ON position.ID = users.positionID ");
    }
}

```

```

        return $statement->fetch_all(MYSQLI_ASSOC);
    }
    public function getPersonClass(int $id)
    {
        $statement = self::$db->prepare(
            "SELECT classNumberType as numberType, classLetterType as letterType " .
            "FROM persons INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN students ON persons.ID = students.PersonID " .
            "INNER JOIN class ON students.classID = class.ID " .
            "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
            "INNER JOIN terms ON studyperiod.termID = terms.ID " .
            "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
            "INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
            "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
            "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
            "WHERE users.ID = ?; ");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function getStudEvaluationBy(int $personid, int $subject) :array
    {
        $statement = self::$db->prepare(
            "SELECT * " . "FROM evaluation WHERE StudentID = $personid AND SubjectID = $subject " .
            "ORDER BY examdate DESC");
        $statement->execute();
        $result = $statement->get_result()->fetch_all(MYSQLI_ASSOC);
        return $result;
    }
    public function getSubjectBy(int $subject) :array
    {
        $statement = self::$db->prepare(
            "SELECT * " . "FROM subjects WHERE ID = $subject ;");
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function getPersonTeacherByUserId($id)
    {
        $statement = self::$db->prepare("SELECT persons.ID as personID , UserID, FName, Lname, Mname, DOB, gender,
" .
            "PhoneN, egn, teachers.ID as TeacherID, username, email, dateofcreate as userDate, " .
            "position.ID as PositionID, position.type as PositionType " .
            "FROM persons " .
            "INNER JOIN teachers ON persons.ID = teachers.PersonID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN position ON teachers.positionID = position.ID " .
            "WHERE users.ID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function getPersonByUserId(int $id)
    {
        $statement = self::$db->prepare(
            "SELECT username,users.status as tatoos,persons.ID as pid,Fname,Mname,Lname,persons.status as pstatus,
" .
            "DOB,egn, gender, email,dateofcreate, countryName,cityName, streetName, users.status as ustatus, " .
            "PhoneN , address.ID as addressID, positionID, users.ID as userID " .
            "FROM persons INNER JOIN address ON persons.AddressID = address.ID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "WHERE UserID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function getPositiondByUser(int $id)
    {
        $statement = self::$db->prepare("SELECT users.ID as userID, position.type as positionType, position.ID as
positionID " .
            "FROM users INNER JOIN position ON users.positionID = position.ID " .
            "WHERE users.ID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function setParentConection(string $parenttype,int $studentID,int $parentID) :bool
    {
        $statement = self::$db->prepare("INSERT INTO familyconnect (connetType, StudentID, ParentID) VALUES (?, ?,
?)");
        $statement->bind_param("sii", $parenttype, $studentID, $parentID);
        $statement->execute();
    }

```



```

        if ($statement->affected_rows !=1){
            return false;
        }
        else{
            return $statement->affected_rows == 1;
        }
    }
    public function kidProfile(int $id) : array
    {
        $statement = self::$db->query("SELECT s.ID as personStudentID, s.Fname as personStudentFname, " .
            "s.Lname as personStudentLname, s.Mname as personStudentMname, classNumberType as numberType, " .
            "classLetterType as letterType, terms.term as term, termYear.termYear as termyear, s.UserID as userid,
            "familyconnect.ID as famID
            FROM students
            INNER JOIN persons s ON students.PersonID = s.ID
            INNER JOIN familyconnect ON students.ID = familyconnect.StudentID
            INNER JOIN parents ON familyconnect.ParentID = parents.ID
            INNER JOIN persons p ON parents.personID = p.ID
            INNER JOIN class ON students.classID = class.ID
            INNER JOIN classgroup ON class.classgroupID = classgroup.ID
            INNER JOIN classletters ON classgroup.letterID = classletters.ID
            INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID
            INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID
            INNER JOIN terms ON studyperiod.termID = terms.ID
            INNER JOIN termyear ON studyperiod.termYearID = termyear.ID
            WHERE parents.ID = $id");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }
    public function getParent(int $id) : array
    {
        $statement = self::$db->query("SELECT p.Fname as parentFname, p.Lname as parentLname, connetType, " .
            "p.UserID as useridparent
            FROM parents INNER JOIN familyconnect ON parents.ID = familyconnect.ParentID
            INNER JOIN persons p ON parents.personID = p.ID
            INNER JOIN students ON familyconnect.StudentID = students.ID
            INNER JOIN persons s ON students.PersonID = s.ID
            INNER JOIN users ON s.UserID = users.ID
            WHERE users.ID = $id");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }
    public function searchEGN($search) : array
    {
        $statement = self::$db->query(
            "SELECT username,users.status as ustatus,persons.ID as pid,Fname,Mname,Lname,persons.status as
            pstatus, " .
            "DOB,egn, gender, email,dateofcreate, countryName,cityName, streetName, position.ID as posID, " .
            "position.type, users.ID as userid ,students.ID as studentID, parents.ID as parentID, " .
            "teachers.ID as teacherID, students.ID as studID " .
            "FROM persons INNER JOIN address ON persons.AdressID = address.ID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "LEFT JOIN teachers ON persons.ID = teachers.PersonID " .
            "LEFT JOIN students ON persons.ID = students.PersonID " .
            "LEFT JOIN parents ON persons.ID = parents.personID " .
            "LEFT JOIN position ON users.positionID = position.ID " .
            "WHERE egn LIKE '%$search%'");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }
    public function getProfileById(int $id)
    {
        $statement = self::$db->prepare(
            "SELECT username,users.status as tatoos,persons.ID as pid,Fname,Mname,Lname,persons.status as pstatus, " .
            "DOB,egn, gender, email,dateofcreate, countryName,cityName, streetName, position.ID as posID, " .
            "position.type as ptype, users.status as ustatus, students.ID as studentID, parents.ID as parentID, " .
            "teachers.ID as teacherID , students.classID as SClassID,formmaster.ID as FMID, formmaster.ClassID as
            FClassID, " .
            "formmaster.TeacherID as FTeacherID, teaching.ID as TTeachingID, " .
            "PhoneN , address.ID as addressID, users.ID as userid, familyconnect.connetType as parentType " .
            "FROM persons INNER JOIN address ON persons.AdressID = address.ID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "LEFT JOIN teachers ON persons.ID = teachers.PersonID " .
            "LEFT JOIN students ON persons.ID = students.PersonID " .
            "LEFT JOIN parents ON persons.ID = parents.personID " .
            "LEFT JOIN familyconnect ON parents.ID = familyconnect.ParentID " .
            "LEFT JOIN position ON users.positionID = position.ID " .
            "LEFT JOIN class ON students.classID = class.ID " .
            "LEFT JOIN formmaster ON class.formMasterID = formmaster.ID " .
            "LEFT JOIN teaching ON class.ID = teaching.ClassID " .
            "WHERE UserID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function createProfile($userId, $fname, $mname, $lname, $dateofbirth, $egn, $AddressID, $gender,

```



```

$phoneN) :bool
{
    $stmCreateProfile = self::$db->prepare(
        "INSERT INTO persons (ID, UserID, Fname, Mname, Lname, DOB, gender, PhoneN, AddressID, egn)
VALUES (NULL, ?, ?, ?, ?, ?, ?, ?, ?, ?);";
    $stmCreateProfile->bind_param("issssssis", $userId, $fname, $mname, $lname, $dateofbirth, $gender,
$phoneN, $AddressID, $egn);
    $stmCreateProfile->execute();
    if($stmCreateProfile->affected_rows !=1){
        return false;
    }else{
        $ProfileID = self::$db->query("SELECT LAST_INSERT_ID()")->fetch_row()[0];
        $_SESSION['profileID'] = $ProfileID;
        $profileStatus = self::$db->query("SELECT status FROM persons WHERE ID = $ProfileID");
        $_SESSION['isreq'] = $profileStatus;
        return $stmCreateProfile->affected_rows == 1;
    }
}

public function createAddress($country, $city, $street) : bool
{
    $q = self::$db->prepare("SELECT ID, countryName FROM countries WHERE countryName = ?");
    $q->bind_param("s", $country);
    $q->execute();
    $result = $q->get_result()->fetch_assoc();
    var_dump($result);
    if($result['countryName'] != $country){
        $stmInsertCountry = self::$db->prepare("INSERT INTO countries (countryName) VALUES (?)");
        $stmInsertCountry->bind_param("s", $country);
        $stmInsertCountry->execute();
        $insertIdCountry = self::$db->insert_id;
    }
    else{
        $insertIdCountry = $result['ID'];
    }
    $qq = self::$db->prepare("SELECT ID, cityName FROM cities WHERE cityName = ?");
    $qq->bind_param("s", $city);
    $qq->execute();
    $resultt = $qq->get_result()->fetch_assoc();
    var_dump($resultt);
    if($resultt['cityName'] != $city){
        $stmInsertCity = self::$db->prepare("INSERT INTO cities (cityName) VALUES (?)");
        $stmInsertCity->bind_param("s", $city);
        $stmInsertCity->execute();
        $insertIdCity = self::$db->insert_id;
    }
    else{
        $insertIdCity = $resultt['ID'];
    }
    $qqq = self::$db->prepare("SELECT ID, streetName FROM streets WHERE streetName = ?");
    $qqq->bind_param("s", $street);
    $qqq->execute();
    $resultttt = $qqq->get_result()->fetch_assoc();
    var_dump($resultttt);
    if($resultttt['streetName'] != $street){
        $stmInsertStreet = self::$db->prepare("INSERT INTO streets (streetName) VALUES (?)");
        $stmInsertStreet->bind_param("s", $street);
        $stmInsertStreet->execute();
        $insertIdStreet = self::$db->insert_id;
    }
    else{
        $insertIdStreet = $resultttt['ID'];
    }
    $stmInsertAddress = self::$db->prepare("INSERT INTO address (countryID, cityID, streetID) " .
        "VALUES (?, ?, ?)");
    $stmInsertAddress->bind_param("iii", $insertIdCountry, $insertIdCity, $insertIdStreet);
    $stmInsertAddress->execute();
    if($stmInsertAddress->affected_rows !=1){
        return false;
    }else{
        $AddressID = self::$db->query("SELECT LAST_INSERT_ID()")->fetch_row()[0];
        $_SESSION['addressID'] = $AddressID;
        return $stmInsertAddress->affected_rows == 1;
    }
}

public function editAddress($country, $city, $street, $addressID) : bool
{
    $q = self::$db->prepare("SELECT ID, countryName FROM countries WHERE countryName = ?");
    $q->bind_param("s", $country);
    $q->execute();
    $result = $q->get_result()->fetch_assoc();

    if($result['countryName'] != $country){
        $stmInsertCountry = self::$db->prepare("INSERT INTO countries (countryName) VALUES (?)");
        $stmInsertCountry->bind_param("s", $country);
        $stmInsertCountry->execute();
        $insertIdCountry = self::$db->insert_id;
    }
    else{
        $insertIdCountry = $result['ID'];
    }
}

```

```

    $qq = self::$db->prepare("SELECT ID, cityName FROM cities WHERE cityName = ?");
    $qq->bind_param("s", $city);
    $qq->execute();
    $resultt = $qq->get_result()->fetch_assoc();
    if($resultt['cityName'] != $city){
        $stmInsertCity = self::$db->prepare("INSERT INTO cities (cityName) VALUES (?)");
        $stmInsertCity->bind_param("s", $city);
        $stmInsertCity->execute();
        $insertIdCity = self::$db->insert_id;
    }
    else{
        $insertIdCity = $resultt['ID'];
    }
    $qqq = self::$db->prepare("SELECT ID, streetName FROM streets WHERE streetName = ?");
    $qqq->bind_param("s", $street);
    $qqq->execute();
    $resultttt = $qqq->get_result()->fetch_assoc();
    if($resultttt['streetName'] != $street){
        $stmInsertStreet = self::$db->prepare("INSERT INTO streets (streetName) VALUES (?)");
        $stmInsertStreet->bind_param("s", $street);
        $stmInsertStreet->execute();
        $insertIdStreet = self::$db->insert_id;
    }
    else{
        $insertIdStreet = $resultttt['ID'];
    }
    $stmUpdateAddress = self::$db->prepare("UPDATE address SET " .
        " countryID = ?, cityID = ?, streetID= ? WHERE ID = ?");
    $stmUpdateAddress->bind_param("iiii", $insertIdCountry, $insertIdCity, $insertIdStreet, $addressID);
    $stmUpdateAddress->execute();
    return $stmUpdateAddress->affected_rows >= 0;
}

public function getClassGroup() : array
{
    $statement = self::$db->query("SELECT classgroup.ID as ID,classLetterType as letter, classNumberType as
number " .
    "FROM classgroup INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
    "INNER JOIN classletters ON classgroup.letterID = classletters.ID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getClsGrpStpBy(int $personsId)
{
    $statement = self::$db->prepare(
        "SELECT classgroup.ID as classgroupID, studyperiod.ID as studyperiodID " .
        "FROM
            students INNER JOIN persons ON students.PersonID = persons.ID
            INNER JOIN users ON persons.UserID = users.ID
            INNER JOIN class ON students.classID = class.ID
            INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID
            INNER JOIN terms ON studyperiod.termID = terms.ID
            INNER JOIN termyear ON studyperiod.termYearID = termyear.ID
            INNER JOIN classgroup ON class.classgroupID = classgroup.ID
            INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID
            INNER JOIN classletters ON classgroup.letterID = classletters.ID
        WHERE persons.ID = ?");
    $statement->bind_param("i", $personsId);
    $statement->execute();
    $result = $statement->get_result()->fetch_assoc();
    return $result;
}

public function editProfile($fname, $mname, $lname, $dateofbirth, $segn, $gender, $phoneN, $profileID) : bool
{
    $stmCreateProfile = self::$db->prepare(
        "UPDATE persons SET Fname = ?, Mname = ? , Lname = ? , DOB = ? , " .
        "gender = ? , PhoneN = ? , egn = ? WHERE ID = ?");
    $stmCreateProfile->bind_param("ssssssi", $fname, $mname, $lname, $dateofbirth, $gender, $phoneN, $segn,
$profileID);
    $stmCreateProfile->execute();
    if($stmCreateProfile->affected_rows !=1){
        return false;
    }else{
        return $stmCreateProfile->affected_rows == 1;
    }
}

public function setPosition($positionid, $userid) : bool
{
    $statement = self::$db->prepare("UPDATE users SET positionID = ? WHERE users.ID = ?");
    $statement->bind_param("ii", $positionid, $userid);
    $statement->execute();
    return $statement->affected_rows == 1;
}

public function getSubjects() : array
{
    $statement = self::$db->query("SELECT * FROM subjects");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getAvailableClass()
{
    $statement = self::$db->query("SELECT class.ID as ClassID ,classNumberType,classLetterType, term, termYear
" .
    "FROM class LEFT JOIN classgroup ON class.classgroupID = classgroup.ID " .

```

```

        "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getStudyPeriod() : array
{
    $statement = self::$db->query(
        "SELECT studyperiod.ID ,term, termYear " .
        "FROM studyperiod INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "ORDER BY termyear.termYear ASC ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function setTeach($teacherID, $subjectset) : bool
{
    $statement = self::$db->prepare("INSERT INTO teach (TeacherID, SubjectID) VALUES ( ? , ?)");
    $statement->bind_param("ii", $teacherID, $subjectset);
    $statement->execute();
    if ($statement->affected_rows !=1){
        return false;
    }
    else{
        return $statement->affected_rows == 1;
    }
}

public function setTeaching($teacherID, $classInsertID) : bool
{
    $statement = self::$db->prepare("INSERT INTO teaching (TeacherID, ClassID) VALUES ( ? , ?)");
    $statement->bind_param("ii", $teacherID, $classInsertID);
    $statement->execute();
    if ($statement->affected_rows !=1){
        return false;
    }
    else{
        return $statement->affected_rows == 1;
    }
}

public function getSubjectsTeacher($teacherID)
{
    $statement = self::$db->query("SELECT subjects.ID as SubID, teach.ID as teachID,Name, ABB " .
        "FROM teach INNER JOIN subjects ON teach.SubjectID = subjects.ID " .
        "WHERE TeacherID = $teacherID");

    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getFormMaster($teacherID)
{
    $statement = self::$db->query("SELECT class.ID as ClassID, term, termYear, classNumberType as number, " .
        " classLetterType as letter " .
        "FROM class INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "LEFT JOIN formmaster ON class.formMasterID = formmaster.ID " .
        "WHERE TeacherID = $teacherID ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getStudentTeachersClass($teacherID)
{
    $statement = self::$db->query("SELECT students.ID as StudentID, Fname, Lname, studentnumber " .
        "FROM class INNER JOIN students ON class.ID = students.classID " .
        "INNER JOIN formmaster ON class.formMasterID = formmaster.ID " .
        "LEFT JOIN persons ON students.PersonID = persons.ID " .
        "WHERE TeacherID = $teacherID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getClassWithoutFormMaster()
{
    $statement = self::$db->query("SELECT class.ID as ClassID, term, termYear, classNumberType,
classLetterType " .
        "FROM class INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "INNER JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "LEFT JOIN formmaster ON class.formMasterID = formmaster.ID " .
        "WHERE class.formMasterID IS NULL ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

```

```

public function getTeachingClass($teacherID)
{
    $statement = self::$db->query("SELECT class.ID as ClassID ,term, termYear, classLetterType as letter,
classNumberType as number, " .
        "teaching.ID as teachingID " .
        "FROM teaching INNER JOIN class ON teaching.ClassID = class.ID " .
        "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "INNER JOIN classnumbers ON classgroup.numberID = classnumbers.ID " .
        "WHERE TeacherID = $teacherID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function setFormMaster($teacherID,$formMasterInsertID) : bool
{
    $stmInsertFormMaster = self::$db->prepare("INSERT INTO formmaster (ClassID,TeacherID) " .
        "VALUES (?, ?)");
    $stmInsertFormMaster->bind_param("ii", $formMasterInsertID, $teacherID);
    $stmInsertFormMaster->execute();
    $stmInsertFormMaster = self::$db->insert_id;
    $statement = self::$db->prepare("UPDATE class SET formMasterID = ? WHERE ID = ? ");
    $statement->bind_param("ii", $stmInsertFormMaster, $formMasterInsertID);
    $statement->execute();
    if($statement->affected_rows !=1){
        return false;
    }else{
        return $statement->affected_rows ==1;
    }
}

public function setClassManager($classManagerInsertID,$ClassManagerID) : bool
{
    $stmInsertClassManager = self::$db->prepare("INSERT INTO classmanagers (StudentID,ClassID) " .
        "VALUES (?, ?)");
    $stmInsertClassManager->bind_param("ii", $classManagerInsertID, $ClassManagerID);
    $stmInsertClassManager->execute();
    $stmInsertClassManager = self::$db->insert_id;
    $statement = self::$db->prepare("UPDATE class SET classManagerID = ? WHERE ID = ? ");
    $statement->bind_param("ii", $stmInsertClassManager, $ClassManagerID);
    $statement->execute();
    if($statement->affected_rows !=1){
        return false;
    }else{
        return $statement->affected_rows ==1;
    }
}

public function getClassManager($teacherID)
{
    $statement = self::$db->query("SELECT persons.ID as pid,Fname, Lname, studentnumber " .
        "FROM classmanagers INNER JOIN students ON classmanagers.StudentID = students.ID " .
        "INNER JOIN persons ON students.PersonID = persons.ID " .
        "INNER JOIN class ON students.classID = class.ID " .
        "INNER JOIN formmaster ON class.formMasterID = formmaster.ID " .
        "INNER JOIN teachers ON formmaster.TeacherID = teachers.ID " .
        "WHERE TeacherID = $teacherID ");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function setClass($classgr,$period)
{
    $stmQuery = self::$db->prepare("SELECT ID FROM class WHERE classgroupID = ? AND studyPeriodID = ?");
    $stmQuery->bind_param("ii",$classgr, $period );
    $stmQuery->execute();
    $classIDInsert = $stmQuery->get result()->fetch row()[0];
    if ($classIDInsert){
        return $classIDInsert;
    }
    else{
        $stmInsertClassData = self::$db->prepare("INSERT INTO class (classgroupID, studyPeriodID) " .
            "VALUES (?, ?)");
        $stmInsertClassData->bind_param("ii", $classgr, $period);
        $stmInsertClassData->execute();
        if($stmInsertClassData->affected_rows !=1){
            return false;
        }else{
            $stmInsertClassData = self::$db->insert_id;
            $_SESSION['classID'] = $stmInsertClassData;
            return $stmInsertClassData;
        }
    }
}

public function setClassStudents ($classgr,$period, $stnumber,$studentid)
{
    $Classid = $this->setClass($classgr,$period);

    $position = $_SESSION['positionID'];
    $stmInsertStudentData = self::$db->prepare("UPDATE students " .
        "SET classID = ? , studentnumber = ? , positionID = ? WHERE PersonID = ?");
    $stmInsertStudentData->bind_param("iiii", $Classid, $stnumber, $position, $studentid);
}

```

```

        $stmInsertStudentData->execute();
        if($stmInsertStudentData->affected_rows !=1){
            return false;
        }else{
            return $stmInsertStudentData->affected_rows == 1;
        }
    }

    public function insertStudent($profileBy,$positionID)
    {
        $stmInsertTeacher = self::$db->prepare("INSERT INTO students (PersonID, positionID) VALUES (?,?)");
        $stmInsertTeacher->bind_param("ii", $positionID,$profileBy);
        $stmInsertTeacher->execute();
        if($stmInsertTeacher->affected_rows !=1){
            return false;
        }else{
            return $stmInsertTeacher->affected_rows == 1;
        }
    }

    public function insertTeacher($profileBy,$positionID)
    {
        $stmInsertTeacher = self::$db->prepare("INSERT INTO teachers ( PersonID, positionID) VALUES (?,?)");
        $stmInsertTeacher->bind_param("ii", $profileBy,$positionID);
        $stmInsertTeacher->execute();
        if($stmInsertTeacher->affected_rows !=1){
            return false;
        }else{
            return $stmInsertTeacher->affected_rows == 1;
        }
    }

    public function insertParent($profileBy,$positionID)
    {
        $stmInsertTeacher = self::$db->prepare("INSERT INTO parents ( personID, positionID) VALUES (?,?)");
        $stmInsertTeacher->bind_param("ii", $positionID,$profileBy);
        $stmInsertTeacher->execute();
        if($stmInsertTeacher->affected_rows !=1){
            return false;
        }else{
            return $stmInsertTeacher->affected_rows == 1;
        }
    }

    public function getClass($PersonID)
    {
        $statement = self::$db->prepare("SELECT class.ID as classID, studyPeriodID, term, termYear, teachers.ID as
teacherID, " .
        "classNumberType as Number, classLetterType as Letter, Fname, Lname, studentnumber " .
        "FROM class INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN teaching ON class.ID = teaching.ClassID " .
        "INNER JOIN teachers ON teaching.TeacherID = teachers.ID " .
        "INNER JOIN formmaster ON class.formMasterID = formmaster.ID " .
        "LEFT JOIN students ON class.ID = students.classID " .
        "LEFT JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "LEFT JOIN classletters ON classgroup.letterID = classletters.ID " .
        "LEFT JOIN claassnumbers ON classgroup.numberID = claassnumbers.ID " .
        "LEFT JOIN persons ON students.PersonID = persons.ID " .
        "WHERE teachers.PersonID = ?");
        $statement->bind_param("i", $PersonID);
        $statement->execute();
        $result = $statement->get_result()->fetch_all(MYSQLI_ASSOC);
        return $result;
    }

    public function deleteTeaching(int $deleteTeaching) : bool
    {
        $statement = self::$db->prepare(
            "DELETE FROM teaching WHERE teaching.ID = ?");
        $statement->bind_param("i", $deleteTeaching);
        $statement->execute();
        return $statement->affected_rows == 1;
    }

    public function deleteSubject(int $deletesubject):bool
    {
        $statement = self::$db->prepare(
            "DELETE FROM teach WHERE teach.ID = ?");
        $statement->bind_param("i", $deletesubject);
        $statement->execute();
        return $statement->affected_rows == 1;
    }

    public function deleteFamConn(int $id) :bool
    {
        $statement = self::$db->prepare(
            "DELETE FROM familyconnect WHERE familyconnect.ID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        return $statement->affected_rows == 1;
    }

    public function delete(int $id) : bool

```

```

{
    $statement = self::$db->prepare(
        "DELETE FROM persons WHERE UserID = ?");
    $statement->bind_param("i", $id);
    $statement->execute();
    return $statement->affected_rows == 1;
}
}

```

UsersModel.php

```
<?php
```

```
class UsersModel extends BaseModel
```

```

{
    public function getSchool() : array
    {
        $statement = self::$db->prepare(
            "SELECT name, countryName, cityName, streetName " .
            "FROM schools INNER JOIN address ON schools.addressID = address.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID " .
            "WHERE schools.ID = 1");
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function login(string $username, string $password)
    {
        $statement = self::$db->prepare(
            "SELECT id, password_hash FROM users WHERE username = ?");
        $statement->bind_param("s", $username);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        if(password_verify($password, $result['password_hash']))
        {
            return $result['id'];
        }
        else return false;
    }
    public function getPositionById(int $id)
    {
        $statement = self::$db->prepare(
            "SELECT users.ID, username, email, positionID,status,type " .
            "FROM users " .
            "INNER JOIN position ON users.positionID = position.ID " .
            "WHERE users.ID = ?");
        $statement->bind_param("i", $id);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }
    public function register(string $username, string $password, string $email)
    {
        $password_hash = password_hash($password, PASSWORD_DEFAULT);
        $statement = self::$db->prepare(
            "INSERT INTO users(username, password_hash, email) VALUES (?, ?, ?)");
        $statement->bind_param("sss", $username, $password_hash, $email);
        $statement->execute();
        if($statement->affected_rows !=1)
            return false;
        $userid = self::$db->query("SELECT LAST_INSERT_ID()")->fetch_row()[0];
        return $userid;
    }
    public function getAllTeachers() : array
    {
        $statement = self::$db->query(
            "SELECT users.ID as UserID, persons.ID as PersonID, teachers.ID as TeacherID, Fname, Lname, PhoneN, " .
            "cities.cityName as city,username, position.ID as positionID ,position.type as positionType " .
            "FROM teachers INNER JOIN persons ON teachers.PersonID = persons.ID " .
            "INNER JOIN users ON persons.UserID = users.ID " .
            "INNER JOIN position ON teachers.positionID = position.ID " .
            "INNER JOIN address ON persons.AddressID = address.ID " .
            "INNER JOIN cities ON address.cityID = cities.ID " .
            "INNER JOIN countries ON address.countryID = countries.ID " .
            "INNER JOIN streets ON address.streetID = streets.ID ");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }
    public function getUserBy(int $personid) : array
    {
        $statement = self::$db->query("SELECT users.ID as UserID, username " .

```

```

        "FROM users LEFT JOIN persons ON users.ID = persons.UserID " .
        "LEFT JOIN teachers ON persons.ID = teachers.PersonID " .
        "LEFT JOIN students ON persons.ID = students.PersonID " .
        "LEFT JOIN parents ON persons.ID = parents.personID " .
        "WHERE persons.ID = $personid ";
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getTecherClassBy(int $teacherid) : array
    {
        $statement = self::$db->query(
            "SELECT terms.term as classterm, termYear as classtermYear, classNumberType, classLetterType " .
            "FROM class " .
            "INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
            "INNER JOIN classnumbers ON classgroup.numberID = classnumbers.ID " .
            "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
            "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
            "INNER JOIN terms ON studyperiod.termID = terms.ID " .
            "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
            "INNER JOIN teaching ON class.ID = teaching.ClassID " .
            "INNER JOIN teachers ON teaching.TeacherID = teachers.ID " .
            "WHERE teachers.ID = $teacherid");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getTecherSubjectBy(int $teacherid) : array
    {
        $statement = self::$db->query("SELECT * " .
            "FROM subjects " .
            "INNER JOIN teach ON subjects.ID = teach.SubjectID " .
            "INNER JOIN teachers ON teach.TeacherID = teachers.ID " .
            "WHERE teachers.ID = $teacherid");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getSubjects() : array
    {
        $statement = self::$db->query("SELECT * FROM subjects");
        return $statement->fetch_all(MYSQLI_ASSOC);
    }

    public function getSubjectBy(int $subject) :array
    {
        $statement = self::$db->prepare(
            "SELECT * " . "FROM subjects WHERE ID = $subject ");
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }

    public function getClsGrpStpBy(int $personsid)
    {
        $statement = self::$db->prepare(
            "SELECT classgroup.ID as classgroupID, studyperiod.ID as studyperiodID " .
            "FROM
                students INNER JOIN persons ON students.PersonID = persons.ID
                INNER JOIN users ON persons.UserID = users.ID
                INNER JOIN class ON students.classID = class.ID
                INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID
                INNER JOIN terms ON studyperiod.termID = terms.ID
                INNER JOIN termyear ON studyperiod.termYearID = termyear.ID
                INNER JOIN classgroup ON class.classgroupID = classgroup.ID
                INNER JOIN classnumbers ON classgroup.numberID = classnumbers.ID
                INNER JOIN classletters ON classgroup.letterID = classletters.ID
            WHERE persons.ID = ?");
        $statement->bind_param("i", $personsid);
        $statement->execute();
        $result = $statement->get_result()->fetch_assoc();
        return $result;
    }

    public function getStudEvaluationBy(int $personid, int $subject) :array
    {
        $statement = self::$db->prepare(
            "SELECT * " . "FROM evaluation WHERE StudentID = $personid AND SubjectID = $subject " .
            "ORDER BY examdate DESC");
        $statement->execute();
        $result = $statement->get_result()->fetch_all(MYSQLI_ASSOC);
        return $result;
    }

    public function getAllStudents(int $classgrp,int $classstp) : array
    {
        $statement = self::$db->query("SELECT students.ID as StudentID, s.ID as PersonID, studentnumber,
classNumberType as numberType, " .
            "classLetterType as letterType, terms.term as term, termyear.termYear as termYear, s.Fname, s.Mname,
s.Lname, " .
            "s.DOB, s.egn, s.PhoneN, countryName, cityName, streetName, position.type as positionType,
formmaster.TeacherID as mssterTeacherID, " .
            "classmanagers.StudentID as managerStudentID, t.Fname as teacherFname, t.Lname as teacherLname,
username, " .
            "users.ID as UserID, position.ID as PositionID " .
            "FROM students INNER JOIN persons s ON students.PersonID = s.ID " .
            "INNER JOIN users ON s.UserID = users.ID " .
            "INNER JOIN position ON students.positionID = position.ID " .
            "INNER JOIN address ON s.AdressID = address.ID " .

```



```

        "INNER JOIN countries ON address.countryID = countries.ID " .
        "INNER JOIN cities ON address.cityID = cities.ID " .
        "INNER JOIN streets ON address.streetID = streets.ID " .
        "INNER JOIN class ON students.classID = class.ID " .
        "INNER JOIN studyperiod ON class.studyPeriodID = studyperiod.ID " .
        "INNER JOIN terms ON studyperiod.termID = terms.ID " .
        "INNER JOIN termyear ON studyperiod.termYearID = termyear.ID " .
        "INNER JOIN classgroup ON class.classgroupID = classgroup.ID " .
        "INNER JOIN classnumbers ON classgroup.numberID = classnumbers.ID " .
        "INNER JOIN classletters ON classgroup.letterID = classletters.ID " .
        "LEFT JOIN formmaster ON class.formMasterID = formmaster.ID " .
        "LEFT JOIN teachers ON formmaster.TeacherID = teachers.ID " .
        "left JOIN persons t ON teachers.PersonID = t.ID " .
        "LEFT JOIN classmanagers ON class.classManagerID = classmanagers.ID " .
        "WHERE classgroup.ID = $classgrp and studyperiod.ID = $classstp;");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getAllParents() : array
{
    $statement = self::$db->query("SELECT Fname, Mname, Lname, DOB, gender, PhoneN, egn, " .
        "position.type as positionType, countryName, cityName, streetName " .
        "FROM parents INNER JOIN persons ON parents.personID = persons.ID " .
        "INNER JOIN position ON parents.positionID = position.ID " .
        "INNER JOIN address ON persons.AddressID = address.ID " .
        "INNER JOIN countries ON address.countryID = countries.ID " .
        "INNER JOIN cities ON address.cityID = cities.ID " .
        "INNER JOIN streets ON address.streetID = streets.ID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getProfileById(int $id)
{
    $statement = self::$db->prepare(
        "SELECT persons.ID as pid, UserID, Fname, Mname, Lname, persons.status as pstatus, username,
positionID, users.status as ustatus " .
        "FROM persons INNER JOIN address ON persons.AddressID = address.ID " .
        "INNER JOIN users ON persons.UserID = users.ID " .
        "INNER JOIN countries ON address.countryID = countries.ID " .
        "INNER JOIN cities ON address.cityID = cities.ID " .
        "INNER JOIN streets ON address.streetID = streets.ID " .
        "INNER JOIN position ON position.ID = users.positionID " .
        "WHERE UserID = ?");
    $statement->bind_param("i", $id);
    $statement->execute();
    $result = $statement->get_result()->fetch_assoc();
    return $result;
}

public function setPosition($profileID,$position)
{
    $statement = self::$db->query("SELECT ID FROM $position WHERE PersonID = $profileID");
    return $statement->fetch_row()[0];
}

public function getPosition() : array
{
    $statement = self::$db->query("SELECT * FROM position");
    return $statement->fetch_all(MYSQLI_ASSOC);
}

public function getAll() : array
{
    $statement = self::$db->query("SELECT *
FROM persons INNER JOIN address ON persons.AddressID = address.ID
INNER JOIN users ON persons.UserID = users.ID
INNER JOIN countries ON address.countryID = countries.ID
INNER JOIN cities ON address.cityID = cities.ID
INNER JOIN streets ON address.streetID = streets.ID
INNER JOIN position ON position.ID = users.positionID");
    return $statement->fetch_all(MYSQLI_ASSOC);
}
}

```

Header.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

```



```

<head>
<!--<link rel="stylesheet" href="<?=APP_ROOT?>/content/css/styles.css" /> -->
<!--<link rel="icon" href="<?=APP_ROOT?>/content/images/favicon.ico" /> -->
<meta http-equiv="content-type" content="text/html; charset=windows-1251" />
<title><?php if (isset($this->title)) echo htmlspecialchars($this->title) ?></title>
<!-- Bootstrap Core CSS -->
<link href="<?=APP_ROOT?>/content/css/bootstrap.min.css" rel="stylesheet">
<link href="<?=APP_ROOT?>/content/css/datepicker.css" rel="stylesheet">
<script src="<?=APP_ROOT?>/content/js/jquery.js"></script>
<script src="<?=APP_ROOT?>/content/js/bootstrap-datepicker.js"></script>
<script>
$(function() {
    $(''.datepicker').datepicker();
});
</script>
<!-- Custom CSS -->
<link href="<?=APP_ROOT?>/content/css/school-admin.css" rel="stylesheet">
<!-- Morris Charts CSS -->
<link href="<?=APP_ROOT?>/content/plugins/morris.css" rel="stylesheet">
<!-- Custom Fonts -->
<link href="<?=APP_ROOT?>/content/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
<!-- Meta for video
<meta name="viewport" content="width=device-width, initial-scale=1">
<!--<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>-->

</head>
<body>
<div id="wrapper">
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
<!-- <a href="<?=APP_ROOT?>"></a> -->
<!-- Brand and toggle get grouped for better mobile display -->
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-
collapse">

<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="<?=APP_ROOT?>/">Начало</a>
</div>
<!-- Top Menu Items -->
<ul class="nav navbar-right top-nav">
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown"></a>
<?php if(!$this->isLoggedIn) : ?>
<i class="fa fa-user"></i> Избери <b class="caret"></b></a>
<?php endif; ?>
<?php if($this->isLoggedIn) : ?>
<i class="fa fa-user"></i> Здравей, <?=htmlspecialchars($_SESSION['username']) ?> <b
class="caret"></b></a>
<?php endif; ?>
<ul class="dropdown-menu">
<?php if(!$this->isLoggedIn) : { ?>
<li>
<a href="<?=APP_ROOT?>/users/login">Вход</a>
</li>
<li>
<a href="<?=APP_ROOT?>/users/register">Регистрация</a>
</li>
<?php } else: {?>
<li>
<a href="<?=APP_ROOT?>/users/logout"><i class="fa fa-fw fa-power-off"></i> Излез</a>
</li>
<?php } endif; ?>
</ul>
</li>
</ul>
<!-- Sidebar Menu Items - These collapse to the responsive navigation menu on
small screens -->
<div class="collapse navbar-collapse navbar-ex1-collapse">
<ul class="nav navbar-nav side-nav">
<li>
<a href="<?=APP_ROOT?>/"><i class="fa fa-fw fa-home"></i>Начална страница</a>
</li>
<?php
if($this->isLoggedIn || isset($_SESSION['profileID'])) :
    $profileid = $_SESSION['profileID'];
    if ($profileid != 0) {
        ?>
<li>
<a href="<?=APP_ROOT?>/profile/create"><i class="fa fa-fw fa-bar-chart-
o"></i>~Създай профил~</a>
</li>
<?php
}
?>
<li>
<a href="<?=APP_ROOT?>/profile"><i class="fa fa-fw fa-barcode"></i>~Профил~</a>

```

```

        </li>
        <?php
        if (isset($_SESSION['studentID'])) {
            $studentID = $_SESSION['studentID'];
        }
        if (isset($_SESSION['teacherID'])) {
            $teacherID = $_SESSION['teacherID'];
        }
        if (isset($_SESSION['parentID'])) {
            $parentID = $_SESSION['parentID'];
        }
        $position = $_SESSION['positionID'];
        if ($position == 1) : ?>
            <li>
                <a href="<?=APP_ROOT?>/notifications"><i class="fa fa-fw fa-
barcode"></i>~Известия~</a>
            </li>
            <?php endif;
        if ($position == 2) :
            ?>
            <li>
                <a href="<?=APP_ROOT?>/class"><i class="fa fa-fw fa-coffee"></i>Клас</a>
            </li>
            <li>
                <a href="<?=APP_ROOT?>/users"><i class="fa fa-fw fa-group"></i>~Потребители~</a>
            </li>
            <li>
                <a href="<?= APP_ROOT ?>/profile/studentProfile/<?= $studentID?>"><i
                    class="fa fa-fw fa-barcode"></i>~Ученически профил~</a>
            </li>
            <?php endif;
        if ($position == 3) :
            ?>
            <li>
                <a href="<?=APP_ROOT?>/events"><i class="fa fa-fw fa-coffee"></i>Статии</a>
            </li>
            <li>
                <a href="<?=APP_ROOT?>/events/create"><i class="fa fa-fw fa-newspaper-
o"></i>Създаване на статия</a>
            </li>
            <li>
                <a href="<?=APP_ROOT?>/class/create"><i class="fa fa-fw fa-coffee"></i>Създай
Клас</a>
            </li>
            <li>
                <a href="<?=APP_ROOT?>/class"><i class="fa fa-fw fa-coffee"></i>Клас</a>
            </li>
            <li>
                <a href="<?=APP_ROOT?>/users"><i class="fa fa-fw fa-group"></i>~Потребители~</a>
            </li>
            <li>
                <a href="<?=APP_ROOT?>/profile/teacherProfile/<?= $teacherID?>"><i
                    class="fa fa-fw fa-barcode"></i>~Учителски профил~</a>
            </li>
            <?php endif;
        if ($position == 4) :
            ?>
            <li>
                <a href="<?= APP_ROOT ?>/profile/parentProfile/<?= $parentID ?>"><i
                    class="fa fa-fw fa-barcode"></i>~Родителски профил~</a>
            </li>
            <?php endif;
        else: {
        }
    endif;
    ?>
</ul>
</div>
</nav>
<div id="page-wrapper">
<!-- </div> -->
<?php require_once('show-notify-messages.php'); ?>

```

View/class/create.php

```

<?php $this->title = 'Създай клас'; ?>
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-md-8">
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h1 class="panel-title"> <?=htmlentities($this->title) ?></h1>
                </div>
                <div class="panel-body">

```

```

<form role="form" method="post">
    <div class="form-inline">
        <label for="term">спок:</label>
        <input id="term" type="text" name="term" class="form-control">
        <label for="termyear">учебна година:</label>
        <input id="termyear" type="text" name="termyear" class="form-control">
        <label for="availablePeriod">
            <select name="availablePeriod" class="form-control">
                <option value="" > налични периоди</option>
                <?php foreach ($this->studyperiod/*formmaster model*/ as $term) :?>
                    <option value="<?=>htmlspecialchars($term['studyPeriodID'])?> "
class="dropdown-item" >
                        <?=>htmlspecialchars($term['term'])?> -
                        <?=>htmlspecialchars($term['termYear'])?> </option>
                <?php endforeach; ?>
            </select>
        </label>
        <button type="submit" value="Create" class="btn btn-default">Изпрати</button>
    </div>
</form>
<form role="form" method="post">
    <div class="form-inline">
        <label for="number">клас от 1 до 12</label>
        <input id="number" type="text" name="number" class="form-control">
        <label for="letter">подгрупа:</label>
        <input id="letter" type="text" name="letter" class="form-control">
        <label for="availableClass">
            <select class="form-control">
                <option value="" > налични класове:</option>
                <?php foreach ($this->classgroup as $classg) :?>
                    <option value="" class="dropdown-item" >
                        <?=>htmlspecialchars($classg['numberType'])?>-
                        <?=>htmlspecialchars($classg['letterType'])?>
                    </option>
                <?php endforeach; ?>
            </select>
        </label>
        <button type="submit" value="Create" class="btn btn-default">Изпрати</button>
    </div>
</form>
<form role="form" method="post">
    <div class="form-inline">
        <label for="subject">въведи предмет:</label>
        <input id="subject" type="text" name="subject" class="form-control">
        <label for="abb">Абривиатура на предмета:</label>
        <input id="abb" type="text" name="abb" class="form-control">
        <label for="availableSubjects">
            <select class="form-control">
                <option value="" > налични предмети</option>
                <?php foreach ($this->subjects as $subject) :?>
                    <option value="<?=>htmlspecialchars($subject['ID'])?> " class="dropdown-
item" >
                        <?=>htmlspecialchars($subject['Name'])?> -
                        <?=>htmlspecialchars($subject['ABB'])?> </option>
                <?php endforeach; ?>
            </select>
        </label>
        <button type="submit" value="Create" class="btn btn-default">Изпрати</button>
    </div>
</form>
</div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title"> <?=> htmlspecialchars($this->title) ?> </h3>
        </div>
        <div class="panel-body">
            <p class="text-center"> <?=> htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left"> Държава: <?=> htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center"> >град/село: <?=> htmlspecialchars($this->school['cityName']) ?></p>
            <p class="text-right">улица: <?=> htmlspecialchars($this->school['streetName']) ?></p>
        </div>
    </div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Най-скорошни събития</h3>
            <!-- s edin foreach wvimamem kvoto ni e nujno -->
        </div>
    </div>
</div>
</div>
<!-- /.container-fluid -->

```

View/class/index.php

```
<?php $this->title = 'Класове'; ?>

<!--<div id="page-wrapper">-->
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-md-12">
            <div class="navbar navbar-inverse">
                <div class="container">
                    <div class="navbar-header">
                        <a class="navbar-brand" href="<?= APP_ROOT ?>/class">Класове</a>
                    </div>
                    <div class="navbar-collapse collapse">
                        <ul class="nav navbar-nav">
                            <?php
                                if (isset($_GET['availablePeriod'])) {
                                    $period = $_GET['availablePeriod'];
                                    foreach ($this->classByNumber as $classNumber) {
                                        if (!isset($currentClass) || $currentClass != $classNumber['classnumbersID']) {
                                            $currentClass = $classNumber['classnumbersID'];
                                            $currentNumber = $classNumber['classNumberType']; ?>
                                        <li class="dropdown">
                                            <a href="<?= APP_ROOT ?>/class/index/?class=<?=$currentClass?>" class="dropdown-
toggle"
                                data-toggle="dropdown"> <?= htmlspecialchars($currentNumber) ?> клас </a>
                                <ul class="dropdown-menu">
                                    <?php
                                        foreach ($this->classByNumber as $classLetter) {
                                            if ($currentClass == $classLetter['classnumbersID']) {
                                                $currentLetter = $classLetter['classID'];
                                                $classLetter = $classLetter['letterType'];
                                                ?>
                                            <li>
                                                <a href="<?= APP_ROOT
?>/class/index/?classletter=<?=$currentLetter?>&period=<?=$period?>">ноября <?= htmlspecialchars($classLetter) ?>
</a>
                                                </li>
                                            <?php } ?>
                                        <?php } ?>
                                    </ul>
                                <?php } } ?>
                            </li>
                        </ul>
                    <ul class="nav navbar-nav">
                        <li>
                            <form action="<?=APP_ROOT?>/class/index/?availablePeriod=" role="form"
method="get">
                                <div class="form-inline">
                                    <label for="availablePeriod">
                                        <select name="availablePeriod" class="form-control">
                                            <option value=""> > налични периоди</option>
                                            <?php foreach ($this->studyperiod as $term) : ?>
                                                <option value="<?=$term['studyPeriodID'] ?>" class="dropdown-
item" >
                                                    <?=htmlspecialchars($term['term']) ?> -
                                                    <?=htmlspecialchars($term['termYear']) ?> </option>
                                                <?php endforeach; ?>
                                            </select>
                                        </label>
                                        <button type="submit" value="Create" class="btn btn-
default">Изпрати</button>
                                    </div>
                                </form>
                            </li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="panel panel-yellow">
        <div class="panel-heading">
        </div>
        <div class="panel-body">
            <?php if ((isset($_GET['classletter']) && isset($_GET['period'])) || isset($_GET['sub'])) {
                ?>
            <div class="col-lg-12">
                <div class="table-responsive">
                    <table class="table table-responsive">
                        <thead>
                            <tr>
                                <th>име</th>
                                <th>№</th>
                                <th>оценка по</th>
                                <th>оценка</th>
                                <th>присъствие по</th>
                                <th>бр.</th>
                            </tr>
```

[illegible]

```

        </a>
    </td>
    <td>
        <?htmlspecialchars($studentid['studentnumber'])?>
    </td>
    <td>
        <a
href="<?=$APP_ROOT?>/class/index/?classletter=<?=$clasgrp?>&period=<?=$period?>" >
        <?htmlspecialchars($this->subjects['Name'])?></a>
    </td>
    <?php
    if (!isset($_GET['evaluation']) && !isset($_GET['edit'])) {
        <td>
            <form method="get">
                <label for="evaluation">
                    <label class="hidden">
                        <input name="student" id="student"
value="<?=$studentGET?>" checked type="radio">
                    </label>
                    <label class="hidden">
                        <input name="subject" id="subject"
value="<?=$subjectGET?>" checked type="radio">
                    </label>
                    <select name="evaluation" class="form-control">
                        <?php
                        foreach ($this->studentmarks as $studentMarks) :
                            $studentMarkID = $studentMarks['ID'];
                            $mark = $studentMarks['exammark'];
                            <?
                            <option value="<?=$studentMarkID?>">
                                <?htmlspecialchars($studentMarks['exammark'])?>-
                        <?htmlspecialchars($studentMarks['examdate'])?></option>
                        <?php
                        endforeach;
                        <?
                    </select>
                    <label class="hidden">
                        <input name="edit" id="edit" value="" checked
type="radio">
                    </label>
                    </label>
                    <button type="submit" value="index" class="btn btn-
default">промени</button>
                </form>
            </td>
            <form method="post">
                <td><label for="evaluation">
                    <label class="hidden">
                        <input name="student" id="student"
value="<?=$studentGET?>" checked type="radio">
                    </label>
                    <label class="hidden">
                        <input name="subject" id="subject"
value="<?=$subjectGET?>" checked type="radio">
                    </label>
                    <select name="evaluation" class="form-control">
                        <?php
                        for ($i=0;$i<=10;$i++){
                            <?
                            <option value="<?=$i?>"><?=$i?></option>
                        <?php
                        }
                    </select>
                    </label>
                    <label class="hidden">
                        <input name="teacher" id="edit" value="<?=$TeacherID?>"
checked type="radio">
                    </label>
                    <button type="submit" value="index" class="btn btn-
default">Добави</button>
                </form>
            <td>
                <?htmlspecialchars($this->teacherBySubject['Fname'])?>
            </td>
            <?php
        }
    <?>
    <?php
    if (isset($_GET['evaluation']) && isset($_GET['edit'])) :
        <?>
        <td>
            <?htmlspecialchars($this->examMarkID['exammark'])?>
        </td>
        <form method="post">
            <label class="hidden">
                <input name="MarkIDdelete" id="" value="<?=$_GET['evaluation']?>"
checked

```

```

                                type="radio">
                            </label>
                            <td>
                                <button type="submit" value="index" class="btn-danger">изтрий
                            </button>
                        </form>
                        <form method="post">
                            <label for="editmark">
                                <label class="hidden">
                                    <input name="MarkID" id="student"
value="<?=$_GET['evaluation']?>" checked type="radio">
                                </label>
                                <select name="editmark" class="form-control">
                                    <?php
                                        for ($i=0;$i<=10;$i++){
                                            <?
                                            <option value="<?=$i;?>"><?=$i?></option>
                                            <?php
                                        }
                                    <?
                                </select>
                            </label>
                            <button type="submit" value="index" class="btn-
warning">Промени</button>
                        </form>
                    </td>
                <?php endif;
            endforeach;
        }
    <?
</table>
</div>
</div>
</div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title"> <?=$_htmlspecialchars($this->title) ?> </h3>
        </div>
        <div class="panel-body">
            <p class="text-center"> <?=$_htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left"> Държава: <?=$_htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center"> град/цело: <?=$_htmlspecialchars($this->school['cityName']) ?></p>
            <p class="text-right"> улица: <?=$_htmlspecialchars($this->school['streetName']) ?></p>
        </div>
    </div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Най-скорошни събития</h3>
            <!-- s edin foreach wvimamem kvoto ni e nujno -->
        </div>
    </div>
</div>
</div>
</div>
<!-- /.container-fluid -->

```

View/events/create.php

```

<?php $this->title = 'Създай събитие';?>
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-12">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <div class="panel-title">
                        <h2><?=$_htmlspecialchars($this->title) ?></h2>
                    </div>
                    <li>
                        <i class="fa fa-home"></i> <a href="<?=$_APP_ROOT?>">Начало</a>
                    </li>
                    <li class="active">
                        <i class="fa fa-edit"></i> <?=$_htmlspecialchars($this->title) ?>
                    </li>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-lg-6">
        <div class="panel panel-yellow">
            <div class="panel-body">
                <form role="form" method="post">
                    <div class="form-group">

```

```

        <label for="event_title">Заглавие:</label>
        <input id="event_title" value="Въведете заглавие на статията" type="text"
name="event_title" class="form-control">
        <label for="event_content">Съдържание:</label>
        <textarea id="event_content" rows="13" name="event_content" class="form-
control"></textarea>
        <button type="submit" value="Create" class="btn btn-success">Създай</button>
        <a href="<?=APP_ROOT?>/events">[Начало]</a> </div>
    </form>
</div>
</div>
</div>
</div>
</div>

```

View/events/delete.php

```

<?php $this->title = 'Изтриване на събитие';?>
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-12">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <div class="panel-title">
                        <h1 class="page-header"><?= htmlspecialchars($this->title) ?></h1>
                        <ol class="breadcrumb">
                            <li>
                                <i class="fa fa-home"></i> <a href="<?=APP_ROOT?>">Начало</a>
                            </li>
                            <li>
                                <i class="fa fa-coffee"></i> <a href="<?=APP_ROOT?>/events">[събития]</a>
                            </li>
                            <li class="active">
                                <i class="fa fa-edit"></i> <?= htmlspecialchars($this->title) ?>
                            </li>
                        </ol>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-lg-6">
            <div class="panel panel-yellow">
                <div class="panel-body">
                    <form role="form" method="post">
                        <div class="form-group">
                            <label for="event_title">Заглавие:</label>
                            <input id="event_title" name="event_title" type="text" disabled
                                value="<?= htmlspecialchars($this->event['Title']) ?>" class="form-control">
                            <label for="event_content">Съдържание:</label>
                            <textarea id="event_content" name="event_content" rows="10" class="form-control"
                                disabled><?=htmlspecialchars($this->event['content']) ?></textarea>
                            <label for="dateofcreate">Дата:</label>
                            <input id="dateofcreate" name="dateofcreate" type="text" class="form-control"
                                value="<?=htmlspecialchars($this->event['dateofcreate']) ?>" disabled/>
                            <label for="event_TeacherID">Автор:</label>
                            <input id="event_TeacherID" name="event_TeacherID" type="text" class="form-control"
                                value="<?=htmlspecialchars(($this->event['TeacherID'])) ?>" disabled/>
                            <button type="submit" value="Delete" class="btn btn-default">Изтрий</button>
                            <a href="<?=APP_ROOT?>/events">[откаж]</a> </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

View/events/edit.php

```

<?php $this->title = 'Редактиране на събитие';?>
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-12">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <h1 class="page-header"><?= htmlspecialchars($this->title) ?></h1>
                    <ol class="breadcrumb">
                        <li>
                            <i class="fa fa-home"></i> <a href="<?=APP_ROOT?>">Начало</a>

```



```

        </li>
        <li>
            <i class="fa fa-coffee"></i> <a href="<?=APP_ROOT?>/events">[събития]</a>
        </li>
        <li class="active">
            <i class="fa fa-edit"></i> <?= htmlspecialchars($this->title) ?>
        </li>
    </ol>
</div>
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-6">
        <div class="panel panel-yellow">
            <div class="panel-body">
                <form role="form" method="post">
                    <div class="form-group">
                        <label for="event_title">Заглавие:</label>
                        <input id="event_title" name="event_title" type="text" name="event_title"
                            value="<?= htmlspecialchars($this->event['Title']) ?>" class="form-control">
                        <textarea id="event_content" name="event_content" rows="10" class="form-control">
                            <?=htmlspecialchars($this->event['content']) ?></textarea>
                        <label for="event_date">Дата:</label>
                        <input id="event_date" name="event_date" type="text" class="form-control"
                            value="<?=htmlspecialchars($this->event['dateofcreate']) ?>" />
                        <label for="user_id">Автор:</label>
                        <input id="user_id" name="user_id" type="text" class="form-control"
                            value="<?=htmlspecialchars(($this->event['TeacherID'])) ?>" />
                        <button type="submit" value="Edit" class="btn btn-default">Промени</button>
                        <a href="<?=APP_ROOT?>/events">[отказ]</a>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

View/events/index.php

```

<?php $this->title = 'Събития'; ?>
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-12">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <h1><?= htmlspecialchars($this->title) ?></h1>
                    <ol class="breadcrumb">
                        <li>
                            <i class="fa fa-home"></i> <a href="<?=APP_ROOT?>">Начало</a>
                        </li>
                        <li class="active">
                            <i class="fa fa-edit"></i> <?= htmlspecialchars($this->title) ?>
                        </li>
                    </ol>
                </div>
            </div>
        </div>
    </div>
    <div class="col-lg-12">
        <div class="panel panel-yellow">
            <div class="panel-body">
                <div class="table-bordered">
                    <table class="table table-hover">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>Title</th>
                                <th>Content</th>
                                <th>Date</th>
                                <th>Author ID</th>
                                <th>Action</th>
                            </tr>
                        </thead>
                        <tbody>
                            <?php foreach ($this->events as $event) : ?>
                                <tr>
                                    <th scope="row"><?= $event['ID'] ?></th>
                                    <td><?= htmlspecialchars($event['Title']) ?></td>
                                    <td><?= cutLongText($event['content'],250) ?></td>
                                    <td><?= htmlspecialchars($event['dateofcreate']) ?></td>
                                    <td><?= $event['TeacherID'] ?></td>
                                    <td><a href="<?=APP_ROOT?>/events/edit/<?=$event['ID'] ?>">[Edit]</a>
                                        <a href="<?=APP_ROOT?>/events/delete/<?=$event['ID'] ?>">[Delete]</a></td>
                                </tr>
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

                <?php endforeach ?>
            </tbody>
        </table>
    </div>
</div>
</div>
<a href="<?=APP_ROOT?>/events/create">[Create new event]</a>
</div>
</div>

```

View/home/index.php

```

<?php $this->title = 'Добре дошли :)' ; ?>

<!--<div id="page-wrapper">-->
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-md-8">
            <div class="panel panel-yellow" >
                <?php foreach ($this->events as $event) : ?>
                    <div class="panel-heading">
                        <h1 class="panel-title"> <?=htmlentities($event['Title']) ?></h1>
                    </div>
                    <div class="panel-body">
                        <p>
                            <i>Create on</i>
                            <?=htmlentities($event['dateofcreate']) ?>
                            <i>by</i>
                            <?=htmlentities($event['Fname']) ?></p>
                        <p> <?=$event['content'] ?></p>
                    </div>
                <?php endforeach ?>
            </div>
        </div>
        <div class="col-md-4">
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h3 class="panel-title"> <?= htmlspecialchars($this->title) ?> </h3>
                </div>
                <div class="panel-body">
                    <p class="text-center" ><?= htmlspecialchars($this->school['name']) ?> </p>
                    <p class="text-left" >Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>
                    <p class="text-center" >град/село: <?= htmlspecialchars($this->school['cityName']) ?></p>
                    <p class="text-right" >улица: <?= htmlspecialchars($this->school['streetName']) ?></p>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <h3 class="panel-title">Най-скорошни събития</h3>
                    <?php foreach ($this->eventsSidebar as $event) : ?>
                        <a href="<?=APP_ROOT?>/home/view/<?=$event['ID'] ?>">
                            <div class="panel-body">
                                <?=htmlentities($event['Title']) ?></a>
                                <p><?=htmlentities($event['dateofcreate']) ?></p>
                            </div>
                        <?php endforeach ?>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<!-- /.container-fluid -->

```

View/home/view.php

```

<?php $this->title = 'Събитие';?>

<div class="container-fluid">
    <!-- Page Heading -->

```

```

<div class="row">
  <div class="col-lg-12">
    <h1 class="page-header"><?= htmlspecialchars($this->title) ?></h1>
    <ol class="breadcrumb">
      <li>
        <i class="fa fa-home"></i> <a href="<?=APP_ROOT?>">Начало</a>
      </li>
      <li class="active">
        <i class="fa fa-edit"></i> <?= htmlspecialchars($this->title) ?>
      </li>
    </ol>
  </div>
</div>
<!-- /.row -->
<div class="row">
  <div class="col-lg-12">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <h3 class="panel-title"><?=htmlspecialchars($this->title) ?></h3>
      </div>
      <div class="panel-body">
        <p>
          <i>Създадено на</i>
          <?=htmlentities($this->event['dateofcreate']) ?>
          <i>,чиито автор е</i>
          <?=htmlentities($this->event['Fname']) ?></p>
        <p> <?=$this->event['content'] ?></p>
      </div>
    </div>
  </div>
</div>
<!-- /.row -->
</div>
<!-- /.container-fluid -->

```

View/notifictions/index.php

```

<?php $this->title = 'Всички потребители'; ?>

<!--<div id="page-wrapper">-->
<div class="container-fluid">
  <!-- Page Heading -->
  <div class="row">
    <div class="col-md-8">
      <div class="panel panel-yellow">
        <div class="panel-heading">
          <h3 class="panel-title">Последните 4 регистрирали се потребителя</h3>
        </div>
        <div class="panel-body">
          <table class="table table-bordered" style="font-size: 12px">
            <thead>
              <tr>
                <th>ID</th>
                <th>Потр.Име</th>
                <th>парола</th>
                <th>e-mail</th>
                <th>Потр.Статус</th>
              </tr>
            </thead>
            <tbody>
              <?php
                foreach ($this->userST as $us) :
                  if (! $us['pstatus']){
                    ?>
                    <tr>
                      <td><p><?=htmlspecialchars($us['ID']) ?></p></td>
                      <td><p><?=htmlspecialchars($us['username']) ?></p></td>
                      <td><p><?= cutLongText($us['password_hash'],33) ?></p></td>
                      <td><p><?=htmlspecialchars($us['email']) ?></p></td>
                      <form method="post">
                        <td><label for="ustatus">
                          <select name="ustatus" class="form-control">
                            <option value="0"><?=$us['ustatus'] ?></option>
                            <option value="1">изчакване</option>
                            <option value="2">активиран</option>
                            <option value="3">забранен</option>
                          </select>
                        </label>
                        <label class="hidden">
                          <input name="useridActive" id="useridActive"
value="<?=$us['ID'] ?>" checked type="radio">

```

```

        </label>
        <button type="submit" value="index" class="btn btn-
default">Изпрати</button>
    </td>
</form>
</tr>
<?php } endforeach;?>
</tbody>
</table>
</div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title"> <?= htmlspecialchars($this->title) ?> </h3>
        </div>
        <div class="panel-body">
            <p class="text-center" ><?= htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left" >Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center" >Град/село: <?= htmlspecialchars($this->school['cityName']) ?> </p>
            <p class="text-right">Улица: <?= htmlspecialchars($this->school['streetName']) ?> </p>
        </div>
    </div>
</div>
<div class="col-md-4" >
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Търсене по ЕГН</h3>
            <?php
            if (!isset($_GET['search']) || isset($_GET['search']) ) {
            ?>
            <form action="<?=APP_ROOT?>/notifications/index/?search=<?php$_GET['search']?>" method="get">
                <label for="search">
                    <input name="search" type="text" class="form-control">
                </label>
                <?php }?>
                <button type="submit" value="Create" class="btn btn-default">Намери</button>
            </form>
        </div>
        <div class="panel-body">
            <table class="table table-bordered" style="font-size: 12px">
                <thead>
                    <tr>
                        <th>Три имена</th>
                        <th>Потр. Статус</th>
                        <th>Проф. Статус</th>
                        <th>Позиция</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    if (isset($_GET['search'])) {
                        foreach ($this->search as $sr) :
                    ?>
                    <tr>
                        <td><a href="<?=APP_ROOT?>/profile/view/<?=$sr['userid']?>" />
                            <p><?=htmlspecialchars($sr['Fname']) ?></p>
                            <p><?=htmlspecialchars($sr['Lname']) ?></p>
                            <p><?=htmlspecialchars($sr['Fname']) ?></p></td>
                        <form method="post">
                            <td><label for="ustatus">
                                <select name="ustatus" class="form-control">
                                    <option value="0"><?=$sr['ustatus']?></option>
                                    <option value="1">изчакване</option>
                                    <option value="2">активиран</option>
                                    <option value="3">забранен</option>
                                </select>
                            </label>
                            <label class="hidden">
                                <input name="useridActive" id="useridActive"
value="<?=$sr['userid']?>" checked type="radio">
                            </label>
                            <button type="submit" value="index" class="btn btn-
default">Изпрати</button>
                        </td>
                    </form>
                    <form method="post">
                        <td><label for="pstatus">
                            <select name="pstatus" class="form-control">
                                <option value="0"><?=$sr['pstatus']?></option>
                                <option value="1">изчакване</option>
                                <option value="2">активиран</option>
                                <option value="3">забранен</option>
                            </select>
                        </label>
                        <label class="hidden">
                            <input name="useridActive" id="useridActive" value="<?=$sr['userid']?>"
checked type="radio">
                        </label>
                        <button type="submit" value="index" class="btn btn-

```

```

default">Изпрати</button>

        </td>
    </form>
    <form method="post">
        <td><label for="posID">
            <select name="posID" class="form-control">
                <option value="0"><?=htmlspecialchars($sr['type'])?></option>
                <option value="1">Администратор</option>
                <option value="2">Ученик</option>
                <option value="3">Учител</option>
                <option value="4">Родител</option>
            </select>
        </label>
        <label class="hidden">
            <input name="useridActive" id="useridActive"
value="<?=$sr['userid']?" checked type="radio">
        </label>
        <label class="hidden">
            <input name="profileID" id="profileID" value="<?=$sr['pID']?" checked
type="radio">
        </label>
        <button type="submit" value="index" class="btn btn-

default">Изпрати</button>

        </td>
    </form>
</tr>
<?php endforeach;??>
</tbody>
</table>
</div>
</div>
</div>
<div class="row">
    <div class="col-md-12">
        <div class="panel panel-yellow" >
            <div class="panel-heading">
                <h1 class="panel-title"> <?=htmlentities($this->title)?></h1>
            </div>
            <div class="panel-body" >
                <table class="table table-bordered" style="font-size: 12px">
                    <thead>
                        <tr>
                            <th>Потреб.Име</th>
                            <th>Дата на създаване</th>
                            <th>Потреб.Статус</th>
                            <th>Три имена</th>
                            <th>Проф.Статут</th>
                            <th>Роден</th>
                            <th>Е-майл</th>
                            <th>ЕГН/лк.№</th>
                            <th>Позиция</th>
                            <th>Адрес</th>
                            <th>Пол</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php ??>
                        <?php
if (!isset($_GET['page'])) {
    $page = 0;
}
else{
    $page = $_GET['page'];
}
foreach ($this->profiles as $profile) : ??<tr>
    <td><p>
        <?= htmlspecialchars($profile['username']) ?></p></td>
    <td><p><?= htmlspecialchars($profile['dateofcreate']) ?></p></td>
    <form method="post">
        <td><label for="ustatus">
            <select name="ustatus" class="form-control">
                <option value="0"><?=$profile['ustatus']?></option>
                <option value="1">изчакване</option>
                <option value="2">активиран</option>
                <option value="3">забранен</option>
            </select>
        </label>
        <label class="hidden">
            <input name="useridActive" id="useridActive"
value="<?=$profile['userid']?" checked type="radio">
        </label>
        <button type="submit" value="index" class="btn btn-

default">Изпрати</button>

        </td>
    </form>
    <td><a href="<?=APP_ROOT?>/profile/view/<?=$profile['userid']?>"/>
        <p><?= htmlspecialchars($profile['Fname']) ?>
        <?= htmlspecialchars($profile['Mname']) ; htmlspecialchars($profile['Lname'])
?></p></td>
    <form method="post">

```

```

        <td><label for="pstatus">
            <select name="pstatus" class="form-control">
                <option value="0"><?=$profile['pstatus']?></option>
                <option value="1">изчакване</option>
                <option value="2">активиран</option>
                <option value="3">забранен</option>
            </select>
        </label>
        <label class="hidden">
            <input name="useridActive" id="useridActive"
value="<?=$profile['userid']?>" checked type="radio">
        </label>
        <button type="submit" value="index" class="btn btn-
default">Изпрати</button>
    </td>
</form>
<td><?=$profile['DOB']?></td>
<td><?=$profile['email']?></td>
<td><?=$profile['egn']?></td>
<form method="post">
    <td><label for="posID">
        <select name="posID" class="form-control">
            <option value="0"><?=$profile['type']?></option>
            <option value="1">Администратор</option>
            <option value="2">Ученик</option>
            <option value="3">Учител</option>
            <option value="4">Родител</option>
        </select>
    </label>
    <label class="hidden">
        <input name="useridActive" id="useridActive"
value="<?=$profile['userid']?>" checked type="radio">
    </label>
    <label class="hidden">
        <input name="profileID" id="profileID" value="<?=$profile['pID']?>"
checked type="radio">
    </label>
    <button type="submit" value="index" class="btn btn-
default">Изпрати</button>
    </td>
</form>
<td><?=$profile['countryName']?>
    рп.<?=$profile['cityName']?>
    улица:<?=$profile['streetName']?>
<td><?=$profile['gender']?></td>
</tr>
<?php endforeach;
if ($page > 0){ $p = $page -1; ??
    <a href="<?=$APP_ROOT?>/notifications/index/?page=<?=$p?>" class="-align-
center">BACK</a>

    <?php } $page++;
    $asd = $_SESSION['asd'];
    $ass = $_SESSION['ass'];
    if ($asd<$ass){??
        <a href="<?=$APP_ROOT?>/notifications/index/?page=<?=$page?>" class="-align-
center">NEXT</a>

    <?php }
    ??
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->

```

View/profile/create.php

```

<?php $this->title = 'Създай профил'; ??

<div class="container-fluid" xmlns="http://www.w3.org/1999/html" xmlns="http://www.w3.org/1999/html"
    xmlns="http://www.w3.org/1999/html">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-8 text-center" disabled="">
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h2 class="panel-title">Регистрирани сте в системата, като
<?=$profile['positiontype']?>
                    <br> потребителско име <?=$profile['username']?></h2>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="panel-body" >
    <form role="form" method="post">
        <?php $exstProfile = $this->profileby['pid'];
        if (isset($exstProfile)) :?>
            <fieldset disabled>
                <?php endif;?>
                <div class="row">
                    <div class="col-sm-6 text-left" >
                        <div class="panel panel-yellow" >
                            <div class="panel-body" >
                                <label for="person_fname" >
                                    <input id="person_fname" type="text" name="person_fname" class="form-
control">

                                <i >Първото Ви име е....</i>
                                </label>
                                <label for="person_mname">
                                    <input id="person_mname" type="text" name="person_mname" class="form-
control">

                                <i >Второто Ви име.....</i>
                                </label>
                                <label for="person_lname">
                                    <input id="person_lname" type="text" name="person_lname" class="form-
control">

                                <i >И третото :).....</i>
                                </label>
                                <label for="person_date" ><i >Роден сте на ?...</i>
                                <input id="person_date" class="form-control; datepicker" type="text"
name="person_date" readonly>

                                </label>
                                <label for="egn">
                                    <input id="egn" type="text" name="egn" class="form-control">
                                    <i >ЕГН или номер по лична карта</i>
                                </label>
                            </div>
                        </div>
                    </div>
                    <div class="col-sm-6 text-left">
                        <div class="panel panel-yellow">
                            <div class="panel-body">
                                <label for="country">
                                    <input id="country" type="text" name="country" class="form-control">
                                    <i>От коя държава сте?</i>
                                </label>
                                <label for="city">
                                    <input id="city" type="text" name="city" class="form-control">
                                    <i>Град или село....</i>
                                </label>
                                <label for="street" class="lin">
                                    <input id="street" type="text" name="street" class="form-control">
                                    <i>Адресът, на който живеете</i>
                                </label>
                                <label for="gender">
                                    <select name="gender" class="form-control">
                                        <option value="" class="dropdown-item" ><i>пол</i></option>
                                        <option value="1" class="dropdown-item" >мъж</option>
                                        <option value="2" class="dropdown-item" >жена</option>
                                    </select>
                                </label>
                                <label for="phonenummer">
                                    <input id="phonenummer" type="text" name="phonenummer" class="form-
control">

                                    <i>Телефон за връзка.....</i>
                                </label>
                            </div>
                        </div>
                    </div>
                    <div class="col-sm-12 text-center">
                        <div class="panel panel-yellow">
                            <div class="panel-body">
                                <button type="submit" value="Create" class="btn btn-
default">Изпрати</button>

                                <a href="<?=APP_ROOT?>">[Начало]</a>
                            </div>
                        </div>
                    </div>
                </div>
            </fieldset>
        </form>
    </div>
</div>
<div class="col-lg-4">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title">Профилът на ..... учаш в:</h3>
        </div>
        <div class="panel-body">
            <p class="text-center" >Средно общообразователно училище</p>
            <p class="text-center">"Добър пример"</p>
            <p class="text-left" >град/село:Пловдив</p> <p class="text-right">район:.....</p>
            <p class="text-left">община:Пловдив</p> <p class="text-right">област:Пловдив</p>

```

```

        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

View/profile/delete.php

```
<?php $this->title = 'Създай профил'; ?>
```

```

<div class="container-fluid" xmlns="http://www.w3.org/1999/html" xmlns="http://www.w3.org/1999/html">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-4">
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h3 class="panel-title">Профилът на ..... учач в:</h3>
                </div>
                <div class="panel-body">
                    <p class="text-center" >Средно общообразователно училище</p>
                    <p class="text-center">"Добър пример"</p>
                    <p class="text-left" >град/село:Пловдив</p> <p class="text-right">район:.....</p>
                    <p class="text-left">община:Пловдив</p> <p class="text-right">област:Пловдив</p>
                </div>
            </div>
        </div>
        <div class="col-lg-8 text-center">
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h3 class="panel-title">Данни на ученика </h3>
                </div>
                <div class="panel-body">
                    <form role="form" method="post">
                        <div class="form-inline">
                            <label for="person_fname">Първото Ви име е:</label>
                            <input id="person_fname" type="text" name="person_fname" class="form-control"
disabled>

                            <label for="person_mname">,второто:</label>
                            <input id="person_mname" type="text" name="person_mname" class="form-control"
disabled>

                            <label for="person_lname">и третото:</label>
                            <input id="person_lname" type="text" name="person_lname" class="form-control"
disabled>

                            <label for="person_date" >Рожден ден:</label>
                            <input id="person_date" class="form-control; datepicker" type="text"
name="person_date" readonly>

                            <label for="egn">ЕГН/лк.№:</label>
                            <input id="egn" type="text" name="egn" class="form-control">
                            <label for="country">Държава:</label>
                            <input id="country" type="text" name="country" class="form-control">

```



```

<label for="city">Град/село:</label>
<input id="city" type="text" name="city" class="form-control">
<label for="street" class="lin"> Улица:</label>
<input id="street" type="text" name="street" class="form-control">
<label for="gender">
  <select name="gender">
    <option value="" class="dropdown-item" > пол</option>
    <option value="1" class="dropdown-item" >мъж</option>
    <option value="2" class="dropdown-item" >жена</option>
  </select>
</label>
<label for="position">
  <select name="position">
    <option value="" class="dropdown-item" > тип потребител</option>
    <option value="2" class="dropdown-item" >ученик</option>
    <option value="3" class="dropdown-item" >учител</option>
    <option value="4" class="dropdown-item" >родител</option>
  </select>
</label>
<label for="phonenumber">Телефон за връзка:</label>
<input id="phonenumber" type="text" name="phonenumber" class="form-control">
<button type="submit" value="Delete" class="btn btn-default">Изтрий</button>

  <a href="<?=APP_ROOT?>">[Начало]</a>
</div>
</form>
</div>
</div>
</div>
<div class="row">
  <div class="col-lg-4 text-center">
    <div class="panel panel-yellow" >
      <div class="panel-heading">
        <h3 class="panel-title">Профилна снимка</h3>
      </div>
      <div class="panel-body">
        <a href="<?=APP_ROOT?>/content/profileImages/<?=$this->profileby['Mname'] ?>.jpg">
          
        </a>
      </div>
    </div>
  </div>
  <div class="col-lg-4 text-center">
    <div class="panel panel-yellow" >
      <div class="panel-heading">
        <h3 class="panel-title">Клас</h3>
      </div>
      <div class="panel-body">
      </div>
    </div>
  </div>
  <div class="col-lg-4 text-center">
    <div class="panel panel-yellow" >
      <div class="panel-heading">
        <h3 class="panel-title">Родители</h3>
      </div>
      <div class="panel-body">
      </div>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-lg-4 text-right">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <p></p>
      </div>
      <div class="panel-body">
        <p></p>
      </div>
    </div>
  </div>
  <div class="col-lg-4 text-right">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <p></p>
      </div>
      <div class="panel-body">
        <p></p>
      </div>
    </div>
  </div>
  <div class="col-lg-4 text-right">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <p></p>
      </div>
      <div class="panel-body">
        <p></p>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
</div>

```

View/profile/edit.php

```
<?php $this->title = 'Ученически профил'; ?>
```

```

<div class="container-fluid" xmlns="http://www.w3.org/1999/html" xmlns="http://www.w3.org/1999/html"
    xmlns="http://www.w3.org/1999/html">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-8 text-center" disabled="">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <h2 class="panel-title">Регистрирани сте в системата, като
<?=htmlspecialchars($_SESSION['positiontype']) ?>
                    <br> потребителско име <?=htmlspecialchars($_SESSION['username']) ?></h2>
                </div>
                <div class="panel-body">
                    <form role="form" method="post">

                        <div class="row">
                            <div class="col-sm-6 text-left">
                                <div class="panel panel-yellow">
                                    <div class="panel-body">
                                        <label for="person_fname">
                                            <input id="person_fname" type="text" name="person_fname"
                                                value="<?=htmlspecialchars($this->profileby['Fname']) ?>"
class="form-control">

                                            <i>Първото Ви име е...</i>
                                        </label>
                                        <label for="person_mname">
                                            <input id="person_mname" type="text" name="person_mname"
                                                value="<?=htmlspecialchars($this->profileby['Mname']) ?>"
class="form-control">

                                            <i>Второто Ви име.....</i>
                                        </label>
                                        <label for="person_lname">
                                            <input id="person_lname" type="text" name="person_lname"
                                                value="<?=htmlspecialchars($this->profileby['Lname']) ?>"
class="form-control">

                                            <i>И третото :).....</i>
                                        </label>
                                        <label for="person_date"><i>Роден сте на ?...</i>
                                            <input id="person_date" class="form-control; datepicker" type="text"
                                                value="<?=htmlspecialchars($this->profileby['DOB']) ?>"
name="person_date" readonly>
                                        </label>
                                        <label for="egn">
                                            <input id="egn" type="text" name="egn"
                                                value="<?=htmlspecialchars($this->profileby['egn']) ?>"
class="form-control">

                                            <i>ЕГН или номер по лична карта</i>
                                        </label>
                                    </div>
                                </div>
                            </div>
                            <div class="col-sm-6 text-left">
                                <div class="panel panel-yellow">
                                    <div class="panel-body">
                                        <label for="country">
                                            <input id="country" type="text" name="country"
                                                value="<?=htmlspecialchars($this->profileby['countryName']) ?>"
class="form-control">

                                            <i>От коя държава сте?</i>
                                        </label>
                                        <label for="city">
                                            <input id="city" type="text" name="city"
                                                value="<?=htmlspecialchars($this->profileby['cityName']) ?>"
class="form-control">

                                            <i>Град или село....</i>
                                        </label>
                                        <label for="street" class="lin">
                                            <input id="street" type="text" name="street"
                                                value="<?=htmlspecialchars($this->profileby['streetName']) ?>"
class="form-control">

                                            <i>Адресът, на който живеете</i>

```

```

        </label>
        <label for="gender">
            <select name="gender" class="form-control">
                <option value="" class="dropdown-item">
                    <i><?=htmlspecialchars($this->profileby['gender'])?></i></option>
                <option value="1" class="dropdown-item">мъж</option>
                <option value="2" class="dropdown-item">жена</option>
            </select>
        </label>
        <label for="phonenumber">
            <input id="phonenumber" type="text" name="phonenumber"
                value="<?=htmlspecialchars($this->profileby['PhoneN'])?>"
class="form-control">

            <i>Телефон за връзка.....</i>
        </label>
    </div>
</div>
</div>
<div class="col-sm-12 text-center">
    <div class="panel panel-yellow">
        <div class="panel-body">
            <button type="submit" value="Create" class="btn btn-
default">Изпрати</button>

            <a href="<?=APP_ROOT?>">[Начало]</a>
        </div>
    </div>
</div>
</div>
</form>
</div>
</div>
<div class="col-lg-4">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title">Профилът на ..... учаш в:</h3>
        </div>
        <div class="panel-body">
            <p class="text-center">Средно общообразователно училище</p>
            <p class="text-center">"Добър пример"</p>
            <p class="text-left">град/село:Пловдив</p> <p class="text-right">район:.....</p>
            <p class="text-left">община:Пловдив</p> <p class="text-right">област:Пловдив</p>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-4 text-center">
        <div class="panel panel-yellow" >
            <div class="panel-heading">
                <h3 class="panel-title">Клас</h3>
            </div>
            <div class="panel-body">
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-center">
        <div class="panel panel-yellow" >
            <div class="panel-heading">
                <h3 class="panel-title">Родители</h3>
            </div>
            <div class="panel-body">
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-center">
        <div class="panel panel-yellow" >
            <div class="panel-heading">
                <h3 class="panel-title">Профилна снимка</h3>
            </div>
            <div class="panel-body">
                <form action="..../upload/" method="post" enctype="multipart/form-data">
                    Select image to upload:
                    <input type="file" name="fileToUpload" id="fileToUpload">
                    <input type="submit" value="Upload Image" name="submit">
                </form>
            </div>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
</div>
</div>

```

View/profile/index.php

```

<?php $this->title = 'Профил'; ??

<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-8 text-center">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <h3 class="panel-title">Потребителско име:
                    <?php $username = $this->personprofile['username'] ??
                    <?= htmlspecialchars($this->personprofile['username']) ?? </h3>
                </div>
                <div class="panel-body">
                    <div class="col-lg-4 text-center">
                        <div class="panel panel-yellow" style="width: auto ; height: auto">
                            <div class="panel-heading">
                                <h3 class="panel-title">Профилна снимка</h3>
                            </div>
                            <div class="panel-body" >
                                <a href="<?=APP_ROOT?>/content/profileImages/<?=$username?>.jpg">
                                    
                                </a>
                            </div>
                        </div>
                    </div>
                    <div class="col-lg-4 text-left">
                        <div class="panel-body">
                            <p><b>Трите ви имена:</b><i><?=htmlspecialchars($this->personprofile['Fname']) ??
                                <?=htmlspecialchars($this->personprofile['Mname']) ??
                                <?=htmlspecialchars($this->personprofile['Lname']) ??</i></p>
                            <p><b>ЕГН/Л.К.№:</b><i><?=htmlspecialchars($this->personprofile['egn']) ??</i></p>
                            <p><b>Пол:</b><i><?=htmlspecialchars($this->personprofile['gender']) ??</i></p>
                            <p><b>E-mail</b><i><?=htmlspecialchars($this->personprofile['email']) ??</i></p>
                            <p><b>Дат на създаване на профил:</b>
                                <i><?=htmlspecialchars($this->personprofile['dateofcreate']) ??</i>
                            </p>
                        </div>
                    </div>
                    <div class="col-lg-4 text-right">
                        <div class="panel-body">
                            <p><b>Държава:</b><i><?=htmlspecialchars($this->personprofile['countryName'])
                                <?=htmlspecialchars($this->personprofile['cityName']) ??</i></p>
                            <p><b>Град:</b><i><?=htmlspecialchars($this->personprofile['cityName']) ??</i></p>
                            <p><b>Адрес:</b><i>ул.<?=htmlspecialchars($this->personprofile['streetName'])
                                <?=htmlspecialchars($this->personprofile['PhoneN']) ??</i>
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-lg-4">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <h3 class="panel-title">Профилът на <?= htmlspecialchars($this->personprofile['username']) ??
учащ в:</h3>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="panel-body">
            <p class="text-center" ><?= htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left" >Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center" >Град/село: <?= htmlspecialchars($this->school['cityName']) ?> </p>
            <p class="text-right">Улица: <?= htmlspecialchars($this->school['streetName']) ?> </p>
        </div>
    </div>
</div>
</div>
</div>
<?php
if ($this->personprofile['positionID'] == 2){
    <div class="row">
        <div class="col-lg-4 text-left">
            <div class="panel panel-yellow">
                <div class="panel-heading">
                    <table class="table table">
                        <thead>
                            <th style="width: 25%">Вие сте в </th>
                            <th class="text-center">Провери оценка по:
                                <form method="get" action="<?=APP_ROOT?>/profile/index/">
                                    <button type="submit" value="index" class="btn btn-default">Изпрати</button>
                                </th>
                        </thead>
                        <tbody>
                            <td>
                                <b>
                                    <?=$this->personcls['numberType']?> <sup> <?=$this->personcls['letterType']?> </sup>
                                </b>
                            </td>
                            <td>
                                <?php
                                $this->Subjects = $this->model->getSubjects();
                                if (!isset($_GET['subject']) || isset($_GET['subject'])) {
                                    <?php
                                    <label for="subject">
                                        <select name="subject" class="form-control">
                                            <?php
                                            foreach ($this->Subjects as $subjectinfo) :
                                                $subjectID = $subjectinfo['ID'];
                                                <option
value="<?=$subjectID?>"><?=htmlspecialchars($subjectinfo['Name']) ?></option>
                                            <?php endforeach;?>
                                        </select>
                                    </label>
                                </form>
                                <?php
                                }
                                <?php
                            </td>
                        </tbody>
                    </table>
                </div>
                <div class="panel-body">
                    <table class="table table-bordered">
                        <thead>
                            <th>Предмет</th>
                            <th>Оценка</th>
                        </thead>
                        <tbody>
                            <?php
                            if (isset($_GET['subject'])) {
                                $subjectid = $_GET['subject'];
                                $subject = $this->model->getSubjectBy($subjectid);
                                $personid = $this->personprofile['pid'];
                                $evaluationsubjectID = $subject['ID'];
                                $this->evaluation = $this->model->getStudEvaluationBy($personid, $evaluationsubjectID
);
                                <td>
                                    <?=htmlspecialchars($subject['Name']) ?>
                                </td>
                                <td>
                                    <label for="evaluation">
                                        <select name="evaluation" class="form-control">
                                            <?php
                                            foreach ($this->evaluation as $evaluation) :
                                                <option value=""><?=htmlspecialchars($evaluation['exammark']) ?> or
дата
                                                <?=htmlspecialchars($evaluation['examdate']) ?></option>

```

```
<?php endforeach;??></select>  
        </label>  
    </td>  
<?php }??>  
</tbody>  
</table>  
</tbody>  
  
</div>  
</div>  
<div class="col-lg-4 text-center">  
    <div class="panel panel-yellow">  
        <div class="panel-heading">  
            <b>Роднини</b>  
        </div>  
        <div class="panel-body">  
            <div class="table-responsive">  
                <table class="table table-responsive">  
                    <thead>  
                        <tr>  
                            <th>име</th>  
                            <th>Родител</th>  
                        </tr>  
                    </thead>  
                    <tbody>  
                        <tr>  
                            <?php foreach ($this->parents as $parent) : ??>  
                                <td><a href="?=APP_ROOT?>/profile/view/<?=$parent['useridparent']"?>">  
                                    <?=htmlspecialchars($parent['parentFname']) ?>  
                                    <?=htmlspecialchars($parent['parentLname']) ?>  
                                </a>  
                                <br>  
                                <?=htmlspecialchars($parent['connetType']) ?>  
                            </td>  
                            <?php endforeach;??>  
                        </tr>  
                    </tbody>  
                </table>  
            </div>  
        </div>  
    </div>  
</div>  
  
</div>  
<?php }
```

View/profile/parentProfile.php

```
<?php $this->title = 'Родителски профил за попълване'; ?>

<!--<div id="page-wrapper">-->
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-md-8 text-center" >
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h2 class="panel-title">Регистрирани сте в системата, като
<?=htmlspecialchars($_SESSION['positiontype'])?>
                    <br> потребителско име <?=htmlspecialchars($_SESSION['username'])?></h2>
                </div>
                <div class="panel-body" >
                    <form action="<?=APP_ROOT?>/profile/edit/<?=htmlspecialchars($this->profileby['pid'])?>"
role="form" method="get">
                        <button type="submit" value="Create" class="btn btn-default">Промени</button>
                    </form>
                    <?php
$sexstProfile = $this->profileby['pid'];
if (isset($sexstProfile))
:??
                    <fieldset disabled>
                        <form role="form" method="post">
                            <?php endif;
                            ??
                            <div class="row">
                                <div class="col-sm-6 text-left" >
                                    <div class="panel panel-yellow" >
                                        <div class="panel-body" >
                                            <label for="person_fname" >
                                                <input id="person_fname" type="text" name="person_fname"
value="<?=htmlspecialchars($this->profileby['Fname'])?>"
class="form-control">

```

```

        <i>Първото Ви име е....</i>
        </label>
        <label for="person_mname">
            <input id="person_mname" type="text" name="person_mname"
                value="<?=htmlspecialchars($this->profileby['Mname'])?>"

class="form-control">

            <i>Второ Ви име....</i>
        </label>
        <label for="person_lname">
            <input id="person_lname" type="text" name="person_lname"
                value="<?=htmlspecialchars($this->profileby['Lname'])?>"

class="form-control">

            <i>И третото :).....</i>
        </label>
        <label for="person_date"><i>Роден сте на ?...</i>
            <input id="person_date" class="form-control; datepicker"

type="text"

                value="<?=htmlspecialchars($this->profileby['DOB'])?>"

name="person_date" readonly>

        </label>
        <label for="egn">
            <input id="egn" type="text" name="egn"
                value="<?=htmlspecialchars($this->profileby['egn'])?>"

class="form-control">

            <i>ЕГН или номер по лична карта</i>
        </label>
    </div>
</div>
</div>
<div class="col-sm-6 text-left">
    <div class="panel panel-yellow">
        <div class="panel-body">
            <label for="country">
                <input id="country" type="text" name="country"
                    value="<?=htmlspecialchars($this->

>profileby['countryName'])?>" class="form-control">

                <i>От коя държава сте?</i>
            </label>
            <label for="city">
                <input id="city" type="text" name="city"
                    value="<?=htmlspecialchars($this->profileby['cityName'])?>"

class="form-control">

                <i>Град или село....</i>
            </label>
            <label for="street" class="lin">
                <input id="street" type="text" name="street"
                    value="<?=htmlspecialchars($this->

>profileby['streetName'])?>" class="form-control">

                <i>Адресът, на който живеете</i>
            </label>
            <label for="gender">
                <select name="gender" class="form-control">
                    <option value="" class="dropdown-item"

><i><?=htmlspecialchars($this->profileby['gender'])?></i></option>
                    <option value="1" class="dropdown-item" >мъж</option>
                    <option value="2" class="dropdown-item" >жена</option>
                </select>
            </label>
            <label for="phonenumber">
                <input id="phonenumber" type="text" name="phonenumber"
                    value="<?=htmlspecialchars($this->profileby['PhoneN'])?>"

class="form-control">

                <i>Телефон за връзка.....</i>
            </label>
        </div>
    </div>
</div>
<div class="col-sm-12 text-center">
    <div class="panel panel-yellow">
        <div class="panel-body">
            <button type="submit" value="Create" class="btn btn-

default">Изпрати</button>

            <a href="<?=APP_ROOT?>">[Начало]</a>
        </div>
    </div>
</div>
</div>
</div>
</fieldset>
</form>
<?php ?>
</div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title"><?= htmlspecialchars($this->title) ?> </h3>
        </div>
        <div class="panel-body">
            <p class="text-center"><?= htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left">Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>

```

```

        <p class="text-center" >град/село: <?= htmlspecialchars($this->school['cityName']) ?></p>
        <p class="text-right">улица: <?= htmlspecialchars($this->school['streetName']) ?></p>
    </div>
</div>
</div>
<div class="col-md-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Търсене по ЕГН</h3>
            <?php
                if (!isset($_GET['search']) || isset($_GET['search'])) {
                    <?
                <form action="<?=APP_ROOT?>/profile/parentProfile/<?=$this->profileby['parentID']?>?search=<?php$_GET['search']?>" method="get">
                    <label for="search">
                        <input name="search" type="text" class="form-control">
                    </label>
                    <?php }?>
                    <button type="submit" value="Create" class="btn btn-default">Намери</button>
                </form>
            </div>
            <div class="panel-body">
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th>Три имена</th>
                            <th>Тип родител</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php
                            if (isset($_GET['search'])) {
                                foreach ($this->search as $sr) :
                                    <tr>
                                        <td><a href="<?=APP_ROOT?>/profile/view/<?=$sr['userid']?>">
                                            <p><?=htmlspecialchars($sr['Fname']) ?></p>
                                            <p><?=htmlspecialchars($sr['Lname']) ?></p>
                                            <p><?=htmlspecialchars($sr['Fname']) ?></p>
                                        </a>
                                        <td>
                                            <form method="post">
                                                <td><label for="parentStudentType">
                                                    <select name="parentStudentType" class="form-control">
                                                        <option value="0">избери</option>
                                                        <option value="майка">Вие сте майка</option>
                                                        <option value="баща">Вие сте баща</option>
                                                        <option value="брат">Вие сте брат</option>
                                                        <option value="сестра">Вие сте сестра</option>
                                                    </select>
                                                </label>
                                                <label class="hidden">
                                                    <input name="studentID" id="studentID" value="<?=$sr['studID']?>"
checked type="radio">
                                                </label>
                                                <label class="hidden">
                                                    <input name="parentID" id="parentID" value="<?=$this->profileby['parentID']?>" checked type="radio">
                                                </label>
                                                <button type="submit" value="index" class="btn btn-default">Изпрати</button>
                                            </form>
                                        </td>
                                    </tr>
                                <?php endforeach; }?>
                            </tbody>
                        </table>
                    </div>
                </div>
</div>
</div>
<div class="row">
    <div class="col-md-8">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <h3 class="panel-title">Вие сте <?=$this->profileby['parentType']?> на</h3>
            </div>
            <div class="panel-body">
                <table class="table table-bordered" style="font-size: 12px">
                    <thead>
                        <tr>
                            <th>Три имена</th>
                            <th>Клас</th>
                            <th>Учебен период</th>
                            <th>Предмет</th>
                            <th>Оценки</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php

```



```

        foreach ($this->kids as $kid) :
            ?>
            <tr>
                <td><a href="<?=APP_ROOT?>/profile/view/<?=$kid['userid']?>">
                    <p><?=htmlspecialchars($kid['personStudentFname']) ?></p>
                    <p><?=htmlspecialchars($kid['personStudentMname']) ?></p>
                    <p><?=htmlspecialchars($kid['personStudentLname']) ?></p>
                </td>
                <td>
                    <p><?=htmlspecialchars($kid['numberType']) ?></p>
                    <p><?=htmlspecialchars($kid['letterType']) ?></p>
                </td> <!-- клас -->
                <td>
                    <p><?=htmlspecialchars($kid['term']) ?></p>
                    <p><?=htmlspecialchars($kid['termyear']) ?></p>
                </td> <!-- Учебен период-->
                <td>
                    <?php
                    if (!isset($_GET['subject'])) {
                        <form action="<?=APP_ROOT?>/profile/parentProfile/<?=$this->profileby['parentID']?>?subject=<?php$_GET['subject']?>" method="get">
                            <label for="subject">
                                <select name="subject" class="form-control">
                                    <?php
                                    foreach ($this->Subjects as $subjectinfo) :
                                        $subjectID = $subjectinfo['ID'];
                                        ?>
                                        <option
                                        value="<?=$subjectID?>"><?=htmlspecialchars($subjectinfo['Name']) ?></option>
                                    <?php endforeach;?>
                                    </select>
                                </label>
                                <button type="submit" value="Create" class="btn btn-
                                default">Намери</button>
                            </form>
                        </td>
                        <td>
                            <form method="post">
                                <label class="hidden">
                                    <input name="delFamilyConn" id="delFamilyConn"
                                    value="<?=$kid['famID']?>" checked type="radio">
                                </label>
                                <button type="submit" value="index" class="btn btn-danger">изтрий</button>
                            </form>
                        </td>
                    <?php
                    }
                    if (isset($_GET['subject'])) {
                        $subjectId = $_GET['subject'];
                        $subject = $this->model->getSubjectBy($subjectId);
                        $evaluationSubjectID = $subject['ID'];
                        $studID = $kid['personStudentID'];
                        $this->evaluation = $this->model->getStudEvaluationBy($studID,
                        $evaluationSubjectID);
                    }

                    <?=htmlspecialchars($subject['Name'])?>

                </td>
                <td>
                    <label for="evaluation">
                        <select name="evaluation" class="form-control">
                            <?php
                            foreach ($this->evaluation as $evaluation) :
                                ?>
                                <option
                                value=""><?=htmlspecialchars($evaluation['exammark']) ?> от дата
                                <?=htmlspecialchars($evaluation['examdate']) ?></option>
                            <?php endforeach;?> </select>
                        </label>
                    </td>
                <?php
                }?>
            </tr>
            <?php endforeach;?>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->

```

View/profile/studentProfile.php

[illegible]

```

        </label>
        <label for="phonenumber">
            <input id="phonenumber" type="text" name="phonenumber"
                value="<?=htmlspecialchars($this->profileby['PhoneN'])?>"
class="form-control">
            <i>Телефон за връзка.....</i>
        </label>
    </div>
</div>
<div class="col-sm-12 text-center">
    <div class="panel panel-yellow">
        <div class="panel-body">
            <button type="submit" value="Create" class="btn btn-
default">Изпрати</button>
            <a href="<?=APP_ROOT?>">[Начало]</a>
        </div>
    </div>
</div>
</div>
</fieldset>
</form>
<?php ?>
</div>
</div>
<div class="col-lg-4">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title">Профилът на ..... учаш в:</h3>
        </div>
        <div class="panel-body">
            <p class="text-center" >Средно общообразователно училище</p>
            <p class="text-center">"Добър пример"</p>
            <p class="text-left" >град/село:Пловдив</p> <p class="text-right">район:.....</p>
            <p class="text-left">община:Пловдив</p> <p class="text-right">област:Пловдив</p>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-4 text-center">
        <div class="panel panel-yellow" >
            <div class="panel-heading">
                <h3 class="panel-title">Клас</h3>
            </div>
            <div class="panel-body">
                <?php $checkPosition = $this->profileby['SClassID'];
                if (($checkPosition)) :?>
                <fieldset disabled>
                    <?php endif;?>
                    <form role="form" method="post">
                        <label for="availableClass">
                            <select name="classgroup" class="form-control" >
                                <option value="" > налични класове:</option>
                                <?php foreach ($this->classgroup as $classg) :
                                    $classgrID = $classg['ID'];
                                    ?>
                                    <option value="<?=$classgrID?>" class="dropdown-item" >
                                        <?=htmlspecialchars($classg['number'])?>-<?=$
htmlspecialchars($classg['letter'])?>
                                    </option>
                                <?php endforeach; ?>
                            </select>
                        </label>
                        <label for="availablePeriod">
                            <select name="period" class="form-control">
                                <option value="" > налични периоди</option>
                                <?php foreach ($this->studyperiod as $term) :
                                    $termID = $term['ID']?>
                                    <option value="<?=$termID?>" class="dropdown-item" >
                                        <?=htmlspecialchars($term['term'])?> -
                                        <?=htmlspecialchars($term['termYear'])?> </option>
                                <?php endforeach; ?>
                            </select>
                        </label>
                        <div class="panel-body text-center">
                            <div class="col-sm-6 ">
                                <div class="panel panel-yellow">
                                    <i>№ в клас</i>
                                    <label for="stNumber" class="col-sm-6 ">
                                        <input id="stNumber" type="text" name="stNumber" class="form-control">
                                        <button type="submit" value="Create" class="btn btn-
default">Изпрати</button>
                                    <a href="<?=APP_ROOT?>">[Начало]</a>
                                </div>
                            </div>
                        </div>
                    </fieldset>
                </form>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
</div>
<div class="col-lg-4 text-center">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title">Родители</h3>
        </div>
        <div class="panel-body">
        </div>
    </div>
</div>
<div class="col-lg-4 text-center">
    <div class="panel panel-yellow" >
        <div class="panel-heading">
            <h3 class="panel-title">Профилна снимка</h3>
        </div>
        <div class="panel-body">
            <form method="post" enctype="multipart/form-data">
                Select image to upload:
                <input type="file" name="image" id="image">
                <input type="submit" value="Upload Image" name="upload">
            </form>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
    <div class="col-lg-4 text-right">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <p></p>
            </div>
            <div class="panel-body">
                <p></p>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

Veiw/profile/teacherProfile.php

```

<?php $this->title = 'Учителски профил'; ?>

<div class="container-fluid" xmlns="http://www.w3.org/1999/html" xmlns="http://www.w3.org/1999/html"
    xmlns="http://www.w3.org/1999/html">
    <!-- Page Heading -->
    <div class="row">
        <!-- Профилна информация плюс бутон за редакция-->
        <div class="col-lg-8 text-center" disabled="">
            <div class="panel panel-yellow" >
                <div class="panel-heading">
                    <h2 class="panel-title">Регистрирани сте в системата, като
<?=htmlspecialchars($_SESSION['positiontype']) ?>
                    <br> потребителско име <?=htmlspecialchars($_SESSION['username']) ?></h2>
                </div>
                <div class="panel-body" >
                    <form action="<?=APP_ROOT?>/profile/edit/<?=htmlspecialchars($this->profileby['pid']) ?>"
role="form" method="get">
                        <label class="hidden">
                            <input name="" value="" checked type="radio">
                        </label>
                        <button type="submit" value="Create" class="btn btn-default">Промени</button>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```

$exstProfile = $this->profileby['pid'];
if (isset($exstProfile) && !isset($_GET['editpage']))
:??
<fieldset disabled>
    <form role="form" method="post">
        <?php endif;
        ?>
        <div class="row">
            <div class="col-sm-6 text-left" >
                <div class="panel panel-yellow" >
                    <div class="panel-body" >
                        <label for="person_fname" >
                            <input id="person_fname" type="text" name="person_fname"
                                value="<?htmlspecialchars($this->profileby['Fname'])?>"
class="form-control">
                            <i>Първото Ви име е...</i>
                        </label>
                        <label for="person_mname">
                            <input id="person_mname" type="text" name="person_mname"
                                value="<?htmlspecialchars($this->profileby['Mname'])?>"
class="form-control">
                        <i>Врото Ви име....</i>
                        </label>
                        <label for="person_lname">
                            <input id="person_lname" type="text" name="person_lname"
                                value="<?htmlspecialchars($this->profileby['Lname'])?>"
class="form-control">
                        <i>И третото :).....</i>
                        </label>
                        <label for="person_date" ><i>Роден сте на ?...</i>
                            <input id="person_date" class="form-control; datepicker"
                                value="<?htmlspecialchars($this->profileby['DOB'])?>"
name="person_date" readonly>
                        </label>
                        <label for="egn">
                            <input id="egn" type="text" name="egn"
                                value="<?htmlspecialchars($this->profileby['egn'])?>"
class="form-control">
                        <i>ЕГН или номер по лична карта</i>
                        </label>
                    </div>
                </div>
            </div>
            <div class="col-sm-6 text-left">
                <div class="panel panel-yellow">
                    <div class="panel-body">
                        <label for="country">
                            <input id="country" type="text" name="country"
                                value="<?htmlspecialchars($this->
>profileby['countryName'])?>" class="form-control">
                        <i>От коя държава сте?</i>
                        </label>
                        <label for="city">
                            <input id="city" type="text" name="city"
                                value="<?htmlspecialchars($this->profileby['cityName'])?>"
class="form-control">
                        <i>Град или село...</i>
                        </label>
                        <label for="street" class="lin">
                            <input id="street" type="text" name="street"
                                value="<?htmlspecialchars($this->
>profileby['streetName'])?>" class="form-control">
                        <i>Град или село...</i>
                        </label>
                        <label for="gender">
                            <select name="gender" class="form-control">
                                <option value="value"<?htmlspecialchars($this->
>profileby['gender'])?>" " class="dropdown-item" ><i>пол</i></option>
                                <option value="1" class="dropdown-item" >мъж</option>
                                <option value="2" class="dropdown-item" >жена</option>
                            </select>
                        </label>
                        <label for="phonenumber">
                            <input id="phonenumber" type="text" name="phonenumber"
                                value="<?htmlspecialchars($this->profileby['PhoneN'])?>"
class="form-control">
                        <i>Телефон за връзка.....</i>
                        </label>
                    </div>
                </div>
            </div>
            <div class="col-sm-12 text-center">
                <div class="panel panel-yellow">
                    <div class="panel-body">
                        <button type="submit" value="Create" class="btn btn-
default">Изпрати</button>
                        <a href="<?APP_ROOT?>">[Начало]</a>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

        <th>Съкращение</th>
        <th>Изтрий</th>
    </tr>
</thead>
<tbody>
<?php foreach ($this->subTeacher as $subjectT) : ?>
    <tr>
        <td><?= htmlspecialchars($subjectT['Name']) ?></td>
        <td><p><?= htmlspecialchars($subjectT['ABB']) ?></td>
        <form method="post">
            <td>
                <label class="hidden">
                    <input name="delSubject" id="deleteSubject"
value="<?=$subjectT['teachID']?>" checked type="radio">
                </label>
                <button type="submit" value="index" class="btn btn-danger">изтрий</button>
            </td>
        </form>
    </tr>
<?php endforeach ?>
</tbody>
</table>
<form role="form" method="post">
    <label for="subject">
        <select name="subjectset" class="form-control">
            <option > налични предмети</option>
            <?php foreach ($this->subjects as $subject) : ?>
                <option class="dropdown-item" value="<?=$subject['ID']?>" >
                    <?=htmlspecialchars($subject['Name'])?> -
                    <?=htmlspecialchars($subject['ABB'])?> </option>
            <?php endforeach; ?>
        </select>
    </label>
    <button type="submit" value="Create" class="btn btn-default">Добави</button>
</form>
</div>
</div>
</div>
<!-- преподавате на-->
<div class="col-lg-4 text-center">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Преподавате на:</h3>
        </div>
        <div class="panel-body">
            <table class="table table-striped table-inverse">
                <thead>
                    <tr>
                        <th>Клас</th>
                        <th>За периода</th>
                        <th>Изтрий</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($this->teachingClass as $teachingClass) : ?>
                        <tr>
                            <td><?= htmlspecialchars($teachingClass['number']) ?> -
                                <?= htmlspecialchars($teachingClass['letter']) ?></td>
                            <td><?= htmlspecialchars($teachingClass['term']) ?> -
                                <?= htmlspecialchars($teachingClass['termYear']) ?></td>
                            <form method="post">
                                <td>
                                    <label class="hidden">
                                        <input name="delTeaching" id="deleteTeaching"
value="<?=$teachingClass['teachingID']?>" checked type="radio">
                                    </label>
                                    <button type="submit" value="index" class="btn btn-danger">изтрий</button>
                                </td>
                            </form>
                        </tr>
                    <?php endforeach ?>
                </tbody>
            </table>
            <form role="form" method="post">
                <label for="classInsert">
                    <select name="classInsert" class="form-control">
                        <option > изберете клас</option>
                        <?php foreach ($this->classavailable as $classAvailable) : ?>
                            <option class="dropdown-item" value="<?=$classAvailable['ClassID']?>" >
                                <?=htmlspecialchars($classAvailable['classNumberType'])?> -
                                <?=htmlspecialchars($classAvailable['classLetterType'])?>
                                за периода: <?=htmlspecialchars($classAvailable['term'])?> срок
                                <?=htmlspecialchars($classAvailable['termYear'])?> г.</option>
                        <?php endforeach; ?>
                    </select>
                </label>
                <button type="submit" value="Create" class="btn btn-default">Добави</button>
            </form>
        </div>
    </div>
</div>

```

```

</div>
<!-- отговорник на класа -->
<div class="col-lg-4 text-center">
  <div class="panel panel-yellow">
    <div class="panel-heading">
      <h3 class="panel-title">Изберете отговорник на класа:</h3>
    </div>
    <div class="panel-body">
      <?php if ($this->chosenClassManager):
        foreach ($this->chosenClassManager as $manager): ?>
          <p><i>Отговорник на Вашият клас е <a
href="<?=$APP_ROOT?>/profile/view/<?=$manager['pid']?>"> <?=htmlspecialchars($manager['Fname']) ?>
          <?=htmlspecialchars($manager['Lname']) ?>
          № <?=htmlspecialchars($manager['studentnumber']) ?></a></i></p>

          <?php endforeach; else: ?><p><i>Няма избран отговорник</i></p>
        <?php endif; ?>
      <?php if (! $this->chosenClassManager):?>
        <form role="form" method="post">
          <label class="hidden">
            <input name="CMclassID" value="<?=$classID?>" checked type="radio">
          </label>
          <label for="classManagerInsert">
            <select name="classManagerInsert" class="form-control">
              <option> изберете отговорник от Вашият клас</option>
              <?php foreach ($this->selectClassManager as $manager):?>
                <option class="dropdown-item" value="<?=$manager['StudentID']?>">
                  <?=htmlspecialchars($manager['Fname']) ?> -
                  <?=htmlspecialchars($manager['Lname']) ?>
                  №: <?=htmlspecialchars($manager['studentnumber']) ?></option>
              <?php endforeach; ?>
            </select>
          </label>
          <button type="submit" value="Create" class="btn btn-default">Добави</button>
        </form>
      <?php endif; ?>
    </div>
  </div>
</div>
<!-- Профи;на снимка-->
<div class="col-lg-4 text-center">
  <div class="panel panel-yellow">
    <div class="panel-heading">
      <h3 class="panel-title">Профилна снимка</h3>
    </div>
    <div class="panel-body">
      <form action=" ../upload/" method="post" enctype="multipart/form-data">
        Select image to upload:
        <input type="file" name="fileToUpload" id="fileToUpload">
        <input type="submit" value="Upload Image" name="submit">
      </form>
    </div>
  </div>
</div>
</div>
<div class="row">
  <div class="col-lg-4 text-right">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <p></p>
      </div>
      <div class="panel-body">
        <p></p>
      </div>
    </div>
  </div>
  <div class="col-lg-4 text-right">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <p></p>
      </div>
      <div class="panel-body">
        <p></p>
      </div>
    </div>
  </div>
  <div class="col-lg-4 text-right">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <p></p>
      </div>
      <div class="panel-body">
        <p></p>
      </div>
    </div>
  </div>
</div>
</div>

```


View/profile/view.php

```
<?php $this->title = 'Профил'; ?>

<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <?php
        $proPositionID = $this->profilePosition['positionID'];
        if ($proPositionID == 2){
            ?>
            <div class="col-lg-8 text-center">
                <div class="panel panel-yellow">
                    <div class="panel-heading">
                        <h3 class="panel-title">Данни за ученика с потребителско
                        име: <?= htmlspecialchars($this->personprofile['username']) ?> </h3>
                    </div>
                    <div class="panel-body">
                        <div class="col-lg-4 text-center">
                            <div class="panel panel-yellow" style="width: 90%; height: 90%">
                                <div class="panel-heading">
                                    <h3 class="panel-title">Профилна снимка</h3>
                                </div>
                                <div class="panel-body">
                                    <a href="<?= APP_ROOT ?>/content/profileImages/avatar.jpg">
                                        
                                    </a>
                                </div>
                            </div>
                        </div>
                        <div class="col-lg-4 text-left">
                            <div class="panel-body">
                                <p><b>Трите ви имена:</b><i><?= htmlspecialchars($this->personprofile['Fname']) ?>
                                    <?= htmlspecialchars($this->personprofile['Mname']) ?>
                                    <?= htmlspecialchars($this->personprofile['Lname']) ?></i></p>
                                <p><b>ЕГН/л.к.№:</b><i><?= htmlspecialchars($this->personprofile['egn']) ?></i></p>
                                <p><b>Пол:</b><i><?= htmlspecialchars($this->personprofile['gender']) ?></i></p>
                                <p><b>Е-mail:</b><i><?= htmlspecialchars($this->personprofile['email']) ?></i></p>
                                <p><b>Дат на създаване на
                                    профил:</b><i><?= htmlspecialchars($this->personprofile['dateofcreate'])
                                ?></i></p>
                            </div>
                        </div>
                        <div class="col-lg-4 text-right">
                            <div class="panel-body">
                                <p><b>Държава:</b><i><?= htmlspecialchars($this->personprofile['countryName'])
                                ?></i></p>
                                <p><b>Град:</b><i><?= htmlspecialchars($this->personprofile['cityName']) ?></i></p>
                                <p><b>Адрес:</b><i>ул.<?= htmlspecialchars($this->personprofile['streetName'])
                                ?></i></p>
                                <p><b>Телефон за връзка:</b><i><?= htmlspecialchars($this->personprofile['PhoneN'])
                                ?></i>
                                </p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-lg-4">
                <div class="panel panel-yellow">
                    <div class="panel-heading">
                        <h3 class="panel-title">Профилът на <?= htmlspecialchars($this->personprofile['username']) ?>
                        учаш
                        в:</h3>
                    </div>
                    <div class="panel-body">
                        <p class="text-center"><?= htmlspecialchars($this->school['name']) ?> </p>
                        <p class="text-left">Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>
                        <p class="text-center">град/село: <?= htmlspecialchars($this->school['cityName']) ?></p>
                        <p class="text-right">улица: <?= htmlspecialchars($this->school['streetName']) ?></p>
                    </div>
                </div>
            </div>
        </div>
        <?php
        ?>
    </div>
    <div class="row">
        <?php
        if ($proPositionID == 3){
            ?>
            <div class="col-lg-8 text-center">
                <div class="panel panel-yellow">
                    <div class="panel-heading">
                        <h3 class="panel-title">Данни за ученика с потребителско
                        име: <?= htmlspecialchars($this->personprofile['username']) ?> </h3>
                    </div>
                    <div class="panel-body">
```

```
<div class="col-lg-4 text-center">
    <div class="panel panel-yellow" style="width: 90%; height: 90%">
        <div class="panel-heading">
            <h3 class="panel-title">Профилна снимка</h3>
        </div>
        <div class="panel-body">
            <a href="<?= APP_ROOT ?>/content/profileImages/avatar.jpg">
                
            </a>
        </div>
    </div>
</div>
<div class="col-lg-4 text-left">
    <div class="panel-body">
        <p><b>Трите ви имена:</b><i><?= htmlspecialchars($this->personprofile['Fname']) ?>
            <?= htmlspecialchars($this->personprofile['Mname']) ?>
            <?= htmlspecialchars($this->personprofile['Lname']) ?></i></p>
        <p><b>ЕГН/л.к.№:</b><i><?= htmlspecialchars($this->personprofile['egn']) ?></i></p>
        <p><b>Пол:</b><i><?= htmlspecialchars($this->personprofile['gender']) ?></i></p>
        <p><b>Е-mail</b><i><?= htmlspecialchars($this->personprofile['email']) ?></i></p>
        <p><b>Дат на създаване на профил:</b><i><?= htmlspecialchars($this->personprofile['dateofcreate'])
?></i></p>
    </div>
</div>
<div class="col-lg-4 text-right">
    <div class="panel-body">
        <p><b>Държава:</b><i><?= htmlspecialchars($this->personprofile['countryName'])
?></i></p>
        <p><b>Град:</b><i><?= htmlspecialchars($this->personprofile['cityName']) ?></i></p>
        <p><b>Адрес:</b><i>ул.<?= htmlspecialchars($this->personprofile['streetName'])
?></i></p>
        <p><b>Телефон за връзка</b><i><?= htmlspecialchars($this->personprofile['PhoneN'])
?></i>
        </p>
    </div>
</div>
</div>
<div class="col-lg-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Профилът на <?= htmlspecialchars($this->personprofile['username']) ?>
                учач
                в:</h3>
        </div>
        <div class="panel-body">
            <p class="text-center"><?= htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left">Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center">град/село: <?= htmlspecialchars($this->school['cityName']) ?></p>
            <p class="text-right">улица: <?= htmlspecialchars($this->school['streetName']) ?></p>
        </div>
    </div>
</div>
</div>
}
?>
</div>
```

View/users/index.php

[illegible]

```

        <a href="<?= APP_ROOT ?>/content/profileImages/avatar.jpg">
            
        </a>
    </div>
</div>
<div class="col-lg-4 text-left">
    <div class="panel-body">
        <p><b>Трите ви имена:</b><i><?= htmlspecialchars($this->personprofile['Fname']) ?>
            <?= htmlspecialchars($this->personprofile['Mname']) ?>
            <?= htmlspecialchars($this->personprofile['Lname']) ?></i></p>
        <p><b>ЕГН/л.к.№:</b><i><?= htmlspecialchars($this->personprofile['egn']) ?></i></p>
        <p><b>Пол:</b><i><?= htmlspecialchars($this->personprofile['gender']) ?></i></p>
        <p><b>Е-mail</b><i><?= htmlspecialchars($this->personprofile['email']) ?></i></p>
        <p><b>Дат на създаване на
            профил:</b><i><?= htmlspecialchars($this->personprofile['dateofcreate'])
?></i></p>
    </div>
</div>
<div class="col-lg-4 text-right">
    <div class="panel-body">
        <p><b>Държава:</b><i><?= htmlspecialchars($this->personprofile['countryName'])
?></i></p>
        <p><b>Град:</b><i><?= htmlspecialchars($this->personprofile['cityName']) ?></i></p>
        <p><b>Адрес:</b><i>ул.<?= htmlspecialchars($this->personprofile['streetName'])
?></i></p>
        <p><b>Телефон за връзка</b><i><?= htmlspecialchars($this->personprofile['PhoneN'])
?></i>
    </p>
    </div>
</div>
</div>
<div class="col-lg-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Профилът на <?= htmlspecialchars($this->personprofile['username']) ?>
учащ
                В:</h3>
        </div>
        <div class="panel-body">
            <p class="text-center"><?= htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left">Държава: <?= htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center">град/село: <?= htmlspecialchars($this->school['cityName']) ?></p>
            <p class="text-right">улица: <?= htmlspecialchars($this->school['streetName']) ?></p>
        </div>
    </div>
</div>
<?php
}
?>
</div>
<div class="row">
<?php
if ($proPositionID == 3){
?>
    <div class="col-lg-8 text-center">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <h3 class="panel-title">Данни за ученика с потребителско
име: <?= htmlspecialchars($this->personprofile['username']) ?> </h3>
            </div>
            <div class="panel-body">
                <div class="col-lg-4 text-center">
                    <div class="panel panel-yellow" style="width: 90%; height: 90%">
                        <div class="panel-heading">
                            <h3 class="panel-title">Профилна снимка</h3>
                        </div>
                        <div class="panel-body">
                            <a href="<?= APP_ROOT ?>/content/profileImages/avatar.jpg">
                                
                            </a>
                        </div>
                    </div>
                </div>
                <div class="col-lg-4 text-left">
                    <div class="panel-body">
                        <p><b>Трите ви имена:</b><i><?= htmlspecialchars($this->personprofile['Fname']) ?>
                            <?= htmlspecialchars($this->personprofile['Mname']) ?>
                            <?= htmlspecialchars($this->personprofile['Lname']) ?></i></p>
                        <p><b>ЕГН/л.к.№:</b><i><?= htmlspecialchars($this->personprofile['egn']) ?></i></p>
                        <p><b>Пол:</b><i><?= htmlspecialchars($this->personprofile['gender']) ?></i></p>
                        <p><b>Е-mail</b><i><?= htmlspecialchars($this->personprofile['email']) ?></i></p>
                        <p><b>Дат на създаване на
                            профил:</b><i><?= htmlspecialchars($this->personprofile['dateofcreate'])
?></i></p>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="col-lg-4 text-right">
            <div class="panel-body">
                <p><b>Държава:</b><i><? = htmlspecialchars($this->personprofile['countryName'])
?></i></p>

                <p><b>Град:</b><i><? = htmlspecialchars($this->personprofile['cityName']) ?></i></p>
?></i></p>

                <p><b>Адрес:</b><i>ул.<? = htmlspecialchars($this->personprofile['streetName'])
?></i>

                <p><b>Телефон за връзка</b><i><? = htmlspecialchars($this->personprofile['PhoneN'])
?></i>

            </div>
        </div>
    </div>
</div>
<div class="col-lg-4">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <h3 class="panel-title">Профилът на <? = htmlspecialchars($this->personprofile['username']) ?>
учащ
            в:</h3>
        </div>
        <div class="panel-body">
            <p class="text-center"><? = htmlspecialchars($this->school['name']) ?> </p>
            <p class="text-left">Държава: <? = htmlspecialchars($this->school['countryName']) ?> </p>
            <p class="text-center">град/село: <? = htmlspecialchars($this->school['cityName']) ?></p>
            <p class="text-right">улица: <? = htmlspecialchars($this->school['streetName']) ?></p>
        </div>
    </div>
</div>
<?php
}
?>
</div>

```

View/users/login.php

```

<?php $this->title = 'Вход'; ?>

<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-12">
            <h1 class="page-header"><? = htmlspecialchars($this->title) ?></h1>
            <ol class="breadcrumb">
                <li>
                    <i class="fa fa-home"></i> <a href="<? = APP_ROOT?>">Начало</a>
                </li>
                <li class="active">
                    <i class="fa fa-edit"></i> Вход
                </li>
            </ol>
        </div>
    </div>
    <div class="row">
        <div class="col-lg-6">
            <form role="form" method="post">
                <div class="form-group">
                    <label for="username">Потребителско име:</label>
                    <input id="username" type="text" name="username" class="form-control" autofocus>
                    <p class="help-block">Изписва се с малка буква без разстояние.</p>
                    <label for="password">Парола:</label>
                    <input id="password" type="password" name="password" class="form-control">
                    <p class="help-block">Цифри и букви.</p>
                    <button type="submit" value="Login" class="btn btn-default">Влез</button>
                    <a href="<? = APP_ROOT?>/users/register">[Нямате регистрация?]</a> </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

View/users/register.php

```

<?php $this->title = 'Регистрация'; ?>

<div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">
        <div class="col-lg-12">

```

```

<h1 class="page-header"><?php htmlspecialchars($this->title) ?></h1>
<ol class="breadcrumb">
    <li>
        <i class="fa fa-home"></i> <a href="<?php APP_ROOT?>">Начало</a>
    </li>
    <li class="active">
        <i class="fa fa-edit"></i> Регистрация
    </li>
</ol>
</div>
</div>
<div class="row">
    <div class="col-lg-6">
        <form role="form" method="post">
            <div class="form-group">
                <label for="username">Потребителско име:</label>
                <input id="username" type="text" name="username" autofocus class="form-control">
                <p class="help-block">Изписва се с малка буква без разстояние.</p>
                <label for="password">Парола:</label>
                <input id="password" type="password" name="password" class="form-control">
                <p class="help-block">Цифри и букви.</p>
                <label for="email">email:</label>
                <input id="email" type="text" name="email" class="form-control">
                <p class="help-block">Цифри и букви. всичко, както и @ </p>
                <button type="submit" value="registration" class="btn btn-default">Регистрация</button>
                <a href="<?php APP_ROOT?>/users/login">[Към входа]</a> </div>
            </div>
        </form>
    </div>
</div>
</div>

```

Config.php

```
<?php
```

```

define('APP_ROOT', '/school');
define('DEFAULT_CONTROLLER', 'home');
define('DEFAULT_ACTION', 'index');
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'schooldb');

```

Functions.php

```
<?php
```

```

/**
 * Parse REQUEST_URI to extract request parts: {controller}/{action}/{params}
 * @return array holding: 'controller' (string), 'action' (string) and
 * 'params' (array of strings). When controller / action is not available,
 * returns the default controller 'home' / action 'index'.
 */
function parseRequest() : array
{
    $requestPath = $_SERVER['REQUEST_URI']; // holds /blog/home/index or /home/index
    if (substr($requestPath, 0, strlen(APP_ROOT . '/')) != APP_ROOT . '/') {
        die('APP_ROOT is incorrectly defined in config.php. Use "" or "/blog".');
    }
    $requestPath = substr($requestPath, strlen(APP_ROOT)); // remove APP_ROOT prefix
    $requestParts = explode('/', $requestPath);
    // Extract the controller from {controller}/{action}/{params}
    $controller = DEFAULT_CONTROLLER;
    if (count($requestParts) >= 2 && $requestParts[1] != '') {
        $controller = $requestParts[1];
    }
    // Extract the action from {controller}/{action}/{params}
    $action = DEFAULT_ACTION;
    if (count($requestParts) >= 3 && $requestParts[2] != '') {
        $action = $requestParts[2];
    }
    // Extract the action parameters from {controller}/{action}/{params}
    $params = array_splice($requestParts, 3);
    $requestParsed = [
        'controller' => $controller,
        'action' => $action,
        'params' => $params];
    return $requestParsed;
}
/**

```

```

* Processes the parsed request [{controller}/{action}/{params}]. Creates and
* initializes the requested controller class, invokes the requested action
* and renders its view.
* @param array $requestParsed associative array holding: 'controller' (string),
* 'action' (string) and 'params' (array of strings).
*/
function processRequest(array $requestParsed)
{
    // Create the controller class
    $controller = $requestParsed['controller'];
    $action = $requestParsed['action'];
    $controllerClassName = ucfirst(strtolower($controller)) . 'Controller';
    if (class_exists($controllerClassName)) {
        $controller = new $controllerClassName($controller, $action);
    } else {
        $controllerFileName = "controllers/" . $controllerClassName . '.php';
        die("Cannot find controller '$controller' in class '$controllerFileName'");
    }
    // Invoke the requested action and renders its view (if not rendered)
    if (method_exists($controller, $action)) {
        // Invoke $controller->{$action}($params);
        $params = $requestParsed['params'];
        call_user_func_array(array($controller, $action), $params);
        $controller->renderView();
    } else {
        die("Cannot find action '$action' in controller '$controllerClassName'");
    }
}
/**
* Auto load the controller and model classes from their .php files.
* @param string $class_name <p>the name of the class to load</p>
*/
function __autoload(string $class_name)
{
    if (file_exists("controllers/$class_name.php")) {
        include "controllers/$class_name.php";
    }
    if (file_exists("models/$class_name.php")) {
        include "models/$class_name.php";
    }
}
function cutLongText(string $text, int $maxSize, bool $htmlEscape = true) : string
{
    $append = '';
    if (strlen($text) > $maxSize) {
        $text = substr($text, 10, $maxSize);
        $append = '&hellip;';
    }
    if ($htmlEscape)
        $text = htmlspecialchars($text);
    return $text . $append;
}

```

Index.php

<?php

```

require_once('config.php');
require_once('functions.php');
session_start();
$requestParsed = parseRequest();
processRequest($requestParsed);

```