

**AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ
AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ**

Əlyazması hüququnda

**İNFORMASIYA VƏ TELEKOMMUNİKASIYA TEXNOLOGİYALARI
FAKÜLTƏSİ**

İxtisasın şifrəsi və adı **060509 “Komputer elmləri”**
İxtisaslaşmanın adı: **“İntellektual sistemlər”**

Mühəndis riyaziyyatı və süni intellekt magistrantı
Quliyeva Vüsalə Rəhim qızının
magistr dərəcəsi almaq üçün
“MƏKTƏBLİNİN DOSTU LAYİHƏSİ ÜZRƏ ORTA MƏKTƏBLƏRDƏ
İDARƏDİLMƏ PROBLEMLƏRİNİN İNTELLEKTUAL ANALİZİ”
Mövzusunda

DİSSERTASIYA İŞİ

Kafedra müdiri:

t.e.d., prof N.B.Ağayev

Elmi rəhbər:

r.f.d., dosent Zamanova Ruhiyyə Bülbül qızı

Bakı 2023

MÜNDƏRİCAT

GİRİŞ	3
I FƏSİL. MƏKTƏBLİNİN DOSTU LAYİHƏSİNİN İDARƏETMƏDƏ ROLU	
1.1. “Məktəblinin Dostu” layihəsi nədir ?	4
1.2.Layihənin və məktəbin idarəetmədə vəzifəsi nələrdir?.....	11
II FƏSİL. CƏDVƏLLƏRİN VERİLƏNLƏR BAZASINDA QURULMASI	
2.1. Baza elementləri	16
III FƏSİL. MƏKTƏBLƏRİN YARANAN PROBLEMLƏRİN PYTHONDA ANALİZİ	
3.1. İdarəetmədə yaranan problemlər hansılardır ?	40
3.2. Problemlərin Pythonda analizi	43
NƏTİCƏ.....	63
İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT SİYAHISI.....	64

GİRİŞ

Mövzunun aktuallığı. Gələcək nəsilin düzgün bir fərd kimi yetişməsi üçün sağlam məktəb mühitinin nə dərəcə önəmli olduğu , şagirdlərin gələcəyə hazırlanmasında necə əhəmiyyətli rolu olduğu məlumdur. Məktəb mühitində şagirdlərin təhsil alması ilə yanaşı onun tərbiyə alması da mühümdür. İdarəetmənin düzgün qurulması və idarəetmə problemlərinin həll edilərək aradan qaldırılması təhsildə sağlam inkişafa təkan verir.

Tədqiqatın obyektı və predmeti. Tədqiqat obyektı kimi “Məktəblinin Dostu” layihəsi və məktəb heyəti,təhsil səviyyəsi və standartları.

Tədqiqatın məqsəd və vəzifələri.Orta məktəblərdə təhsil səviyyəsinin yüksəldilməsi.Problemlərin aradan qaldırılması və yüksək səviyyədə idarə olunma.

Dissertasiyanın elmi yeniliyi. Tədqiqat işinin elmi yeniliyinə aşağıdakılar aiddir:

- məktəb təhsil sisteminin idarəedilməsi araşdırılmışdır ;
- burada yaranan problemlər müzakirə edilmişdir ;
- problemlərin python proqramı ilə təsviri verilmişdir ;
- problem həllinin düzgün idarə olunma ilə arasında bağlılıq göstərilmişdir ;
- layihə öhdəlikləri ;

Dissertasiyanın praktik əhəmiyyəti. Dissertasiya işində verilən təklif və tövsiyələrin həyata keçirilməsi , tədris işinin mükəmməlləşdirilməsi tədris müəssisələrində idarəetmənin planlaşdırılmasının tətbiqinə və təkmilləşdirilməsinə müəyyən qədər yardımçı ola bilər.

Dissertasiya işinin strukturu. Dissertasiya girişdən, hər biri bir-iki yarım fəsildən ibarət olan üç fəsildən, nəticədən və bibliografiyadan ibarətdir.

I FƏSİL. MƏKTƏBLİNİN DOSTU LAYİHƏSİNİN TƏRKİBİ HAQQINDA

1.1. “Məktəblinin Dostu” layihəsi nədir ?

“Məktəblinin Dostu” layihəsi Azərbaycanın Elm və Təhsil Nazirliyi və Azərbaycan Respublikası Prezidenti yanında Vətəndaşlara Xidmət və Sosial İnnovasiyalar üzrə Dövlət Agentliyinin yaratdığı birgə layihəsidir. Məktəblər müəllimlər, şagirdlər və digər təhsil işçiləri ilə yanaşı “Məktəblinin Dostlar”ı fəaliyyət göstərir. Layihə 24 dekabr 2016-cı ildə yaradılmışdır. Layihənin əsas məqsədi adından görüldüyü kimi məktəblilərin, şagirdlərin dostu olmaqla yanaşı eyni zamanda məktəb rəhbərliyi ilə də dostluq münasibətləri quraraq məktəbin uğurlu və şəffaf idarə edilməsi, tədris prosesinin sağlam və təhlükəsiz şəkildə təşkilindən ibarətdir. Sözsüz ki, bu amillərə valideynlərin övladlarının təhsil şəraiti və təhlükəsizliyi barədə məlumatlı olması və məmnunluq dərəcəsi də daxil edə bilərik.

”Məktəblinin Dostu” layihəsinin məqsədlərindən biri müəllim və şagirdlər üçün təhlükəsiz mühit yaratmaqla yanaşı, baş verəcək hər hansısa fəvqəladə halda ilkin tibbi yardım göstərməkdir. Eyni zamanda fəvqəladə və tibbi yardımdan başqa psixoloji yardımlarının da böyükdür. Layihə dünyanın bir çox ölkələrində uğur qazanmışdır. Türkiyə, ABŞ, Gürcüstan, Çexiya, Fransa kimi ölkələrdə layihə müxtəlif adlarda aktivdir. Bu layihə bir az gecikmiş layihədir: "Azərbaycanda təhsil müəssisələri ənənəvi idarəetmədən yeni idarəetməyə keçib. Biz bu zamanı görmüşük ki, məktəb idarəçiliyi-direktor, direktor müavini, sinif rəhbərləri vasitəsilə həyata keçirilmişdir. Amma bu sadaladığımız şəxslər birbaşa tədris prosesinin iştirakçısı olduğu üçün şagirdlərlə geniş şəkildə fəaliyyət göstərmək imkanına, onlarla daha yaxından münasibət qurmağa malik deyillər. Sinif rəhbəri və yaxud da direktor müavini aktiv tədris prosesinin iştirakçısı olduğuna görə şagirdlərin bu və ya digər ehtiyacları, asudə vaxtlarının dəyərləndirilməsi ilə bağlı fəaliyyət göstərə bilmirlər, yəni müəllim sinifdə 45 dəqiqə ərzində həm şagirdlərdə dərs keyfiyyətini yoxlayıb, həm yeni dərsi tədris edib, həm də onların hər biri ilə yaxından ünsiyyət qurub, onlarla məşğul olmaq imkanına malik deyillər.

Daha öncələr orta məktəblərdə hər bir məktəbli ilə ayrı-ayrılıqda onların problemləri, sıxıntıları ilə bağlı işlər güclü şəkildə həyata keçirilə bilmirdi. Amma "Məktəblinin dostu"

vəzifəsini yerinə yetirən gənc əməkdaşlarımız tədris prosesinin iştirakçısı olmadığı üçün birbaşa məktəbdə sosial fəaliyyəti həyata keçirmə imkanına malikdirlər.

Bu layihənin işi prinsipi bir növ Azərbaycan Respublikası Prezidenti Yanında Vətəndaşlara Xidmət və Sosial İnnovasiyalar üzrə Dövlət Agentliyinin işinə bənzəyir. Yaxşı tərəfi ondan ibarətdir ki, seçilmiş şəxslər, yəni məktəblinin dostları həm əlavə iş yeri, həm də kifayət qədər təhsilli, aktiv gənclərin təhsilə cəlb olunması deməkdir. Bu gün ürək rahatlığı ilə deyə bilərik ki, bu layihə olduqca uğurlu bir layihədir.

Layihə ilk dəfə başlayarkən 30 məktəb daxilində aktiv idi. Daha sonra əhatə dairəsi genişləndirilərək Bakı, Sumqayıt və Abşeron regionları üzrə 220 məktəb əhatə etməyə başladı. Bunlardan 20-si Sumqayıt, 20-si Abşeron və 180-i isə Bakı şəhərindəki məktəblərdir. 400-ə yaxın əməkdaşın fəaliyyət göstərdiyi bu layihə 400.000 şagirdi əhatə edir. Ötən aylarda 6yaşını tamamlayan bu layihə artıq özünü isbat etmiş, uğurlu layihə olaraq qeydə alınmışdır.

-Bəs bu layihənin gördüyü işlər nələrəndən ibarətdir ?

“Məktəblinin Dostu”nun əsas vəzifəsi başda da qeyd etdiyimiz kimi məktəblinin eyni zamanda məktəb rəhbərliyinin və işçi heyətinin təhlükəsizliyini qorumaq, fəvqəladə hal baş verdikdə müəllim heyətinə və şagirdlərə ilkin yardım göstərməkdən ibarətdir. Sözsüz ki bu, çox yaxşı təşəbbüsdür. Bunun hesabına ilk növbədə məktəblərdə nizam-intizam problemi həll olunur. Eləcə də məktəblərdə müəllimlə şagird arasında müsbət ab-hava yaranır.

Məktəblərdə işləyən bu layihə əməkdaşları əslində tədrisin “görünməyən qəhrəmanlarıdır”. Məktəblərdə tədris zamanı şagirdlər məktəbə gəlməzdən öncə yəni, 07:30 etibar ilə məktəblinin dostlarının fəaliyyəti başlayır və şagirdləri hər səhər qapıda qarşılayan dostlarımızın iş saati günün sonunda ən sonuncu şagird məktəbdən çıxdıqdan sonra bitir. İş prinsipi olaraq hər şagirdin məktəbə daxil olması və hər şagirdin məktəbi tərk etməsi, valideyninə təhvil bu dostlat tərəfindən izlənilir. Valideynləri gecikən şagirdlərin yanında olaraq onları gözdən qoymur, maraqlanır və valideynlərə təhvil verirlər. Hər məktəbdə şagird sayından asılı olaraq minimum 4, maximum isə 8 əməkdaş olur. Hər mərtəbədə ən az bir əməkdaş olmaqla və kamera nəzarətilə həyata keçirilir. Bu əməkdaşlar günün sonuna qədər daima şagirdlərlə birbaşa ünsiyyətdə olub, onlarla qarşılıqlı dostluq əlaqəsində olurlar. Bütünlükdə məktəbi əhatə edən Təhsil Nazirliyinin layihə kameraları

olur və bu kameraları izləyən daima növbəli şəkildə bir əməkdaş mütləq olmalıdır və bu kameralar heç vaxt nəzarətsiz qalmamalıdır. Bəzən məktəblilərarası mübahisələr , davalar olur və bu əksər vaxtlarda dəhlizlərdə , oğlanlar üçün olan sanitari qovşaqlarında baş verir. Kameralara nəzarət vasitəsilə dəhlizlərdə baş verən hadisələr anında nəzarətə alınır və müdaxilə edilir. Boz uniforma ,ağ köynək, üzərində bayraq rənglərindən ibarət Elm və Təhsil Nazirliyinin loqolu qalustuk geyinən dostlar daima dəhlizlərdə gəzib,ara-sıra sanitari qovşaqlarına və hər tənəffüs zamanı bütün siniflərə nəzarət edib, hər hansı problem olduqda dərhal müdaxilə edirlər.

Ən sadə problemlərdən ən böyük problemlərə qədər hər bir məsələyə ciddi , diqqət və şəffafla yanaşan layihə əməkdaşları eyni zamanda məktəbdaxili bütün hər şeyin məxfi qalmasına da böyük əhəmiyyət verirlər. Baş verən hadisələr haqqında birbaşa olaraq onlar tərəfindən məktəb rəhbərliyi və layihə rəhbərliyinə məlumat verilir.

“Məktəblinin Dostları” gün ərzində şagirdlərin mütəmadi içində olub ,onlarla söhbətlər aparır. Müəllimlərin dərslərə vaxtında daxil olması, dərslərin boş keçirilməməsinə və siniflərin boş qalmamasına nəzarət edir. Çünki boş qalmış dərslər şagirdlərin hansısa problem yaratmasına şərait yaradır. Əgər dərs boşdursa dərhal direktor muavininə xəbər edilir,müəllim yoxdursa həmin sinifə əvəzedici göndərilir. Gün ərzində baş verən bütün hadisələr SOFT adlanan sistemə hesabat kimi yazılır və layihə analitikləri tərəfindən gündəlik hesabatlar oxunur Nazirliyə və Bakı Şəhər Təhsil İdarəsinə göndərilir. Və problemlər həll olunana qədər BTŞİ tərəfindən izlənilir.

“Məktəblinin Dostları” şagirdlərin problemlərindən əlavə onların dünyagörüşünü artırmaq, tədrisə qarşı maraq yaratmaq , asudə vaxtlarını doğru dəyərləndirmək və onların bacarıqlarını qabartmaq üçün müxtəlif intellektual , onların dünyagörüşünü artıran və həvəsləndirici yarışlar keçirirlər. Və bu yarış qaliblərinə sertifikatlar , medallar və müxtəlif hədiyyələr verilir. Bu yarışlara şahmat oyunları, bayramlarla bağlı rəsm və maket müsabiqələri , voleybol yarışları,futbol yarışları, məktəblilərarası bilik yarışları və .s daxildir.

Yay aylarında isə şagirdlərin asudə vaxtlarını keçirilməsi üçün layihə olan məktəblərdə Yay məktəbləri təşkil olunur.Yay məktəbləri 2 hissədən ibarət olmaqla – İyul və Avqust aylarında səhər saat 9-dan günorta saat 2-cən davam edir. Burada əsas məqsəd şagirdlərin asudə vaxtları daha maraqlı , əyləncəli və təhlükəsiz keçirilməsidir. Məktəblinin Dostları

burada həm şagirdlərlə bərabər əylənir eyni zamanda onlarla daha yaxın dostluq əlaqəsində olurlar. Bu Yay məktəbləri “Məktəblinin Dostu” layihəsinin təşəbbüsü ilə 5 və 6-cı sinif şagirdləri üçün təşkil edilir. Hər gün himn oxunduqdan sonra səhər idmanı ilə başlayıb, daha sonra müxtəlif əyləncəli oyunlar, cizgi film günləri və başqa intellektual oyunlar ilə davam edirlər. Onu da qeyd edək ki, layihə könüllü formada həyata keçirilir. Şagirdlər öz istəkləri ilə Yay məktəbi üçün ərizələrini verir və 1 aylıq Yay məktəbinə qatılırlar. Təkcə Yay məktəblərində deyil, tədris ili boyunca da şagirdlər üçün müxtəlif oyun, bilik, qabiliyyət yarış və müsabiqələri keçirərək şagirdlərin təhsilə təşəbbüsünə təkan verirlər. Bir neçəsini sadalasaq: Bakı şəhəri 47, 200 və 3 nömrəli tam orta məktəblərin 6-cı sinif şagirdləri arasında futbol üzrə keçirilən yarış, Bakı şəhər 286 nömrəli məktəbdə ibtidai siniflər arasında şahmat üzrə keçirilən yarış, Bakı şəhəri 57 nömrəli tam orta məktəbdə 57, 45 və 200 nömrəli tam orta məktəblərin şagirdləri arasında məktəblərarası keçirilən şahmat yarış, Bakı şəhər 27 nömrəli tam orta məktəbdə tarix fənni üzrə 27, 59, 194 və 17 nömrəli məktəblərarası bilik yarış keçirilib. Bu sadəcə son 1 neçə həftənin göstəricisidir. Şagirdləri ruhlandırان, həvəsləndirən bu yarışların sayı-hesabı yoxdur.

“Məktəblinin Dostları” məktəblərdə şagirdlərə çox vacibi olan təlimləri hansı ki, hadisə baş verdikdə ilkin yardım üçün lazım olan təlimi “İlkin yardım”, hər hansı bədbəxt hadisə zəlzələ, yanğın, uçqun və başqa bu kimi hadisələr baş verdikdə “Fövqəladə hallar üçün təlim”, gələcəkdə ixtisas seçmək üçün “Düzgün ixtisas seçimi təlimləri”, sağlam qidalanmaq üçün qida balans təlimləri keçirirlər. Təkcə bu dostlar şagirdlərə yox eyni zamanda müəllimlər üçün də “Stresin idarə edilməsi” barəsində müxtəlif psixoloji təlimlər keçirirlər.

“Məktəblinin Dostları” qeyd etdiyimiz kimi məktəblilər üçün müəyyən təlimlər keçirirlər. Lakin ilk öncə onlar özləri bu təlimlərə cəlb edilir. Yaxın günlərdə aldığımız məlumat əsasən həmin bu dostlarımıza “Məktəblinin dostu” layihəsi üzrə, Bakı Şəhər Dövlət Yol Polisi İdarəsi və Azərbaycan Milli Avtomobil Klubunun birgə əməkdaşlığı çərçivəsində məktəblinin dostları üçün “Yol hərəkəti qaydalarını uşaqlara necə öyrətməli?” mövzusunda təlimlər həyata keçirilmişdir. “Məktəblinin Dostlarına” bu təlimdə uşaqlara yol hərəkət qaydalarını necə keçmək lazımdır, necə etmək olar ki, şagirdlərin marağını cəmləyərək mənimsədisin bu cür metodlar öyrədilir. Sözsüz ki, ibtidai sinif şagirdlərinə

sırf bunu mühazirə şəklində söyləsək onlar bunu maraqla dinləməyəcək amma onlara bu təlimləri müəyyən oyunlar yaxud real hadisələrlə göstərsən artıq onlar müəyyən qədər biliklərə sahib olacaqlar. Yaxud da 6-10 yaşlı uşaqlar üçün nəzərdə tutulan təlim metodunu 8-11-ci sinif şagirdlərinə tətbiq edilsə istənilən nəticəni almaq mümkün deyil, çünki bu onlara maraqlı gəlməyəcək. Hər bir yaş bölgüsünün özünə xas öyrənmə metodu var.

Eyni zamanda şagirdlərlə dost olmaqla yanaşı onların gündəlik əhvalı, dərsləri ilə mütəmadi maraqlanıb, hər hansı problem olduqda yardım edib, böyük dostları, bacı-qardaşları kimi məsləhət verib onlara yaxın olurlar. Bu dostların bir müsbət cəhəti də ondan ibarətdir ki, məktəblilərə yaş baxımından yaxın olduqları üçün dostluq əlaqələri qurmaqda da çətinlik çəkmirlər. Yəni onları yaxşı anlayır, yaşadıkları çətinlikləri, problemləri dərk edib onlara öz həyatlarında əldə etdikləri təcrübə vasitəsilə köməklik edirlər. Çünki onlar da bir zaman gənc olmuş və bu yollardan keçmiş, istər dərslər baxımından, istərsə də bu yaşda yaşanan sıxıntıları yaşamışlar. Sözsüz ki, onların hər çətinliyini anlamaq, onların hər yaşadığı şeyi yaşamaqdan söz getmir. Çünki onların içərisindən təəssüf ki, bəziləri valideynlərini itirmiş, bəzilərinin valideyni ayrılmış və bu kimi mürəkkəb hallar yaşamış şagirdlərin sayı heç də az deyil. Amma burada əsas məqsəd gənc olan bu dostların – “Məktəblinin Dostları” bu cür situasiyalarla qarşılaşan zaman həmin şagirdə necə yanaşmalı olduğunu bilir, onunla empati qurub, özünü onun yerinə qoyaraq ona məsləhətlər verir və daima onu izləyib yanında olduğunu bildirir.

Şagirdlərin psixologiyasını hər şeydən üstün tutan bu gənclər hər hansı travmatik hadisə baş verdiyi zaman dərhal layihə psixoloqları ilə görüş təşkil edib, şagirdin düzgün psixikada olması üçün əlindən gələni edirlər. Məktəbdə rast gəlinən hadisələr zamanı istər ciddi istər qeyri-ciddi hadisələr zamanı onlar lazımi tədbirləri görürlər. Əgər şagird zədə alıbsa məktəbdə bu hər hansı səbəbdən dərhal ilk iş olaraq təcili tibbi yardıma zəng edilir. Dəfələrlə bu tip hadisələrdə məktəb rəhbərlikləri sırf məsələ böyüməsin, polis məsələyə qarışmasın deyərək bu tip hadisələrin qarşısını almaq üçün 103-ə zəng etməyə tərəddüd etmiş və daha pis nəticələrlə sonlanmışdır. Buna nümunə olaraq 2013-cü ildəki Elena Hacıyeva adlı şagirdin intihar bariz nümunədir. Məktəbli intihar etməzdən öncə, sanki kiminsə onu xilas etməsin gözləyirmişcəsinə dəqiqələrlə pəncərə kənarında qalmışdır. Lakin diqqətsizlik və ya əhəmiyyət verməmək şagirdin özünü atmasına gətirib çıxarmışdır. Yıxıldıqdan sonra belə

ona yardım edilməmiş , müşahidə kameralarına baxdıqda məlum olmuşdur ki, intihar hadisəsindən 2saat sonra təcili tibbi yardıma zəng edilmişdir. Və nəticə olaraq şagird həyatını itirmişdir. Buradan da bu dostların işinə nə qədər şəffaf və nə qədər həssaslıqla yanaşdığı , öncəliyin həqiqətən şagird və müəllim heyətinin sağlamlığı olduğu məlumdur.

“Məktəblinin Dostları” eyni zamanda güvənlə olan şagird deməkdir hər bir valideyn üçün. Bu qədər göstərilən əmək təbii ki, əbəs yerə deyildir. Valideynlərin əksəriyyəti bu dostları sevir və dəstəkləyir. Hətta layihənin olmadığı bir çox məktəblərdə valideynlər və direktorlar bu layihənin olmasını istəyir və bunun üçün layihə rəhbərlərinə müraciət etmişlər. Yaxın zamanda layihənin bir İnstitut şəklində olacağı təsadüfi deyildir. Olduqca uğurlu və faydalı bir layihədir.

“Məktəblinin Dostları” təkcə məktəblərdə fəaliyyət göstərmir , keçdiyimiz yaxın illərdə koronavirus əleyhinə vaksın zamanı könüllü yardım etmiş , son bir neçə ildə müəllimlərin işə qəbul imtahanlarının keçirilməsində Bakı şəhəri və ətrafı, digər bölgələr üzrə yaxından iştirak etmiş bu və digər dövlət tədbirlərində hazırki günə qədər də iştirakları davam edir.

Layihə yarandığı gündən bu günə qədər minlərlə şagirdin, valideynlərin , məktəb rəhbərliyinin sevgisini qazanmışdır. Burada tək bir məqsəd var müəllimin də, layihə əməkdaşının da , valideynin də istəyi şagirdin təhlükəsiz və sağlam şəkildə tədriş alması , gələcəyə sağlam gəncliyin yaradılmasıdır. Sağlam gənclik sağlam cəmiyyət, sağlam gələcək deməkdir.

LAYİHƏ İŞÇİLƏRİ HAQQINDA

Koordinatorlar kimlərdir ? Layihə əməkdaşlarının içərisində müəyyən təcrübəyə sahib olan əməkdaşlar var ki, bu şəxslər koordinator adlanırlar. Koordinatorlar bölgələr üzrə bir neçə məktəbi öhdəçiliyinə götürərək həftə ərzində 2-3 dəfədən az olmayaraq məktəblərə ziyarət edib, oradakı vəziyyətlə maraqlanırlar, baş verən hadisələri müzakirə edirlər , əgər problemlər varsa həllində yaxından iştirak edirlər , əməkdaşlarla söhbətlər və lazımi gəldikdə iclaslar keçirirlər. Əməkdaşlar gündəlik baş verən hadisələri SOFT adlanan hesabatlara daxil edirlər. Koordinatorlar hər gün mütəmadi olaraq bu hesabatları oxuyub məktəblərə nəzarət edirlər.

Analitiklər kimlərdir ? Məktəblinin dostlarının gündəlik SOFT-a daxil etdiyi hesabatları anbaan izləyib, hadisələrin gedişatı ilə maraqlanan , mühüm hesabatları açıq saxlayıblar

hadisənin nəticəsini izləyən əməkdaşlar analitiklərdir. Onlar hər gün hesabatları izləyib , vacib hadisələri qeydə alır və ümumi gündəlik olaraq Bakı Şəhər Təhsil İdarəsinə məlumat verirlər.

Psixoloqlar – layihənin ən mühüm tərkib hissəsi də adlandırılabilir. Layihə psixoloqları həm əməkdaş qəbulun zamanı müsahibələrdə , həm yeni əməkdaşlara psixoloji təlimlərin keçirilməsində eyni zamanda hadisə baş verdikdə məktəblərdə yaxından iştirak edirlər. Psixoloqlar məktəblərdə hər hansısa problem yaşanıbsa və burada şagirdlərdə hər hansı bir psixoloji problem yaradacaq nüanslar varsa dərhal həmin məktəblərə baş çəkib psixoloji söhbətlər aparırlar. Təkcə şagirdlərlə deyil , lazımı gəldikdə əməkdaşlara da böyük yardımlar göstərilir.

1.2. Məktəb idarəetməsi nədir və necə olmalıdır ?

Məktəb heyətində əsas iştirakçı kimlərdir?

Bildiyimiz kimi məktəb təkcə təhsil verən tədris müəssəsi deyil, eyni zamanda təlim və tərbiyə verən , şagirdləri bir fərd kimi formalaşmasında böyük rol oynayan təhsil ocağıdır. Bu prosesin icrasında iştirak edən pedaqoq kollektivinə nəzərə yetirək : məktəb direktoru,təlim-tərbiyə üzrə direktor müavini,təşkilati üzrə direktor müavini,təsərrüfat işləri üzrə direktor müavini, müəllimlər.

Məktəbin idarə edilməsi məktəbin düzgün işləməsini və təhsilin effektiv şəkildə həyata keçirilməsini təmin etmək üçün onun əməliyyatlarına və resurslarına nəzarət və əlaqələndirmə prosesinə aiddir. Bu, müxtəlif aspektləri, o cümlədən idarəetmə, kurikulumun hazırlanması, müəllim nəzarəti, tələbə dəstəyi və ictimaiyyətlə əlaqəni əhatə edir. Məktəb idarəetməsinin bəzi əsas elementləri bunlardır:

Liderlik və İdarəetmə: Məktəbin effektiv idarə edilməsi güclü liderlik və idarəetmə tələb edir. Direktorlar və ya müdirlər kimi məktəb rəhbərləri məktəbin vizyonunu, missiyasını və məqsədlərini təyin etmək, mühüm qərarlar qəbul etmək, işçi heyətini idarə etmək və qaydalara uyğunluğu təmin etmək üçün məsuliyyət daşıyırlar.

Strateji Planlaşdırma: Məktəb rəhbərliyi məktəbin məqsədlərini, strategiyalarını və fəaliyyət planlarını əks etdirən uzunmüddətli strateji planların hazırlanmasını nəzərdə tutur. Bu planlar məktəbin fəaliyyətinə, resursların bölüşdürülməsinə və qərar qəbul etmə proseslərinə rəhbərlik edir.

Kurikulumun hazırlanması və həyata keçirilməsi: Məktəbin idarə edilməsinə təhsil standartlarına uyğun gələn, şagird ehtiyaclarına cavab verən və effektiv tədris və öyrənməni təşviq edən kurikulumun layihələndirilməsi, işlənilib hazırlanması və həyata keçirilməsi daxildir. Bu, müvafiq dərsləklərin və təlim resurslarının seçilməsi, cədvəllərin təşkili və kurikulumun çatdırılmasına nəzarəti əhatə edir. **İşçi heyətinin idarə edilməsi:** Müəllimləri və digər işçiləri idarə etmək məktəb idarəçiliyinin mühüm aspektidir. Buraya müəllimlərin işə qəbulu, işə götürülməsi və təlimi, həmçinin peşəkar inkişaf imkanlarının təmin edilməsi, fəaliyyətinin qiymətləndirilməsi və müsbət iş mühitinin təşviqi daxildir.

Tələbələrə Dəstək və İntizam: Məktəb rəhbərliyi tələbələrin rifahını və akademik uğurlarını təmin etmək üçün onlara dəstək xidmətləri göstərməyi nəzərdə tutur. Bura

konsultasiya, xüsusi təhsil xidmətləri, karyera rəhbərliyi və intizam problemlərinin həlli daxildir. Şagirdlərin inkişafı üçün müsbət və inklüziv məktəb mədəniyyətinin yaradılması vacibdir.

Maliyyə menecmenti: Səmərəli maliyyə idarəçiliyi məktəbin düzgün işləməsi üçün çox vacibdir. Bu, büdcənin tərtibini, resursların bölüşdürülməsini, maliyyə planlamasını və maliyyə məsələlərində şəffaflığın və hesabatlılığın təmin edilməsini əhatə edir.

Valideynlər və İcma İştirakı: Effektiv məktəb idarəçiliyi valideynlər, qəyyumlar və yerli icma ilə güclü münasibətlərin gücləndirilməsini nəzərdə tutur. Maraqlı tərəflərlə əməkdaşlıq və ünsiyyət dəstək əldə etməyə, narahatlıqları həll etməyə və onları məktəb fəaliyyətlərinə və qərarların qəbuluna cəlb etməyə kömək edir.

Məlumatların İdarə Edilməsi və Qiymətləndirilməsi: Məktəb rəhbərliyi şagird performansını, davamiyyət və digər müvafiq göstəricilər haqqında məlumatların toplanması və təhlilini əhatə edir. Məlumata əsaslanan bu yanaşma tərəqqinin monitorinqinə, təkmilləşdirilməli sahələrin müəyyən edilməsinə və əsaslandırılmış qərarların qəbul edilməsinə kömək edir.

Bu göstərilən aspektləri effektiv şəkildə idarə etməklə yanaşı, məktəb rəhbərliyi həmçinin, məktəbin səmərəli işləməsini təmin edir, əlverişli öyrənmə mühiti ortaya çıxarır və şagirdlərə öz potensiallarını tam şəkildə göstərməyə gözəl bir potensial yaradır.

Bu vəzifələri aşağıdakı pillələrə ayırmaq :

- birinci pillə – məktəbin idarəetməsində aparıcı rolu olan – məktəb direktoru;
- ikinci pillə – direktor köməkçisi olan təlim-tərbiyə işləri üzrə müavinlər;
- üçüncü pillə – pedaqoji kollektivin ictimai təşkilatları və kolleqial idarəetmə orqanları;
- dördüncü pillə – məktəbin bütün müəllimləri və sinif rəhbərləri aiddir;
- beşinci pillə – müəllimlər və şagirdlər; xüsusilə, yuxarı sinif şagirdləri, birgə fəaliyyət göstərən orqanlar (pedaqoji və şagird kollektivinin birgə şurası, məktəbdə keçirilən xətt, müxtəlif komissiya və şuraların işi və s.);
- altıncı pillə – şagird kollektivi, gənclər və uşaq birlikləri (ubr), şagird özünüidarə orqanları;
- yeddinci pillə – bu pillədə biz şagird obyektinə kollektivin bir üzvü olaraq baxa bilərik.

Direktor – direktor obrazını biz bir məktəblərdə baş idarəçilər adlandırma bilərik. Direktor məktəb binasındakı bütün müəllimlərə və digər məktəb işçilərinə nəzarət etmək və onları idarə etmək üçün məsuliyyət daşıyır. Direktor vəzifəsini müəllimlərin lideri də adlandırma bilərik. Direktorlar adətən şagirdlərə dərs keçmirlər. Bunun əvəzinə direktor müəllimlərə kömək edir, nizam-intizamda kömək edir, müəyyən qərarlar qəbul edir və məktəbə rəhbərlik edir. Direktor təhsilin strukturunun yenilənib hazırlanması, yeni tədris planının yaradılması və proqramlarının hazırlanması, təlim metodları və qiymətləndirmə kateqoriyalarının işlənilib hazırlanması əsas məzmununa nəzarət edir.

Direktor müavinləri -böyük məktəblərdə direktor müavinləri var. Müdir müavinləri direktora məktəbin idarə olunmasında kömək edir. Müavinləri ümumtəhsil məktəbində idarəedici arasında əsas funksiyalar daşıyanları kimi hesab edə bilərik. İdarəedici yalnız tədris işləri üzrə cavabdeh şəxsdir. Bu zaman həmin vəzifənin tədris işləri üzrə direktor müavinini adlandırmaq daha doğru olardı. Tədris hissəsini idarə edən və eyni zamanda ona nəzarəti təmin edən şəxs təlim – tərbiyə işləri üzrə direktor direktor müavinidir. Sovet dövründəki məktəblərdə bu vəzifə tədris işləri üzrə direktor müavinini adlanırdı. Onda biz qeyd edə bilərik ki, tədris işləri üzrə direktor müavinini, məktəbdə tədrisin təşkilatçısı həm də nəzarətçisi və əlaqələndirici rolunu oynamaqla məktəb kollektivinin əsas aparıcı rollarından biridir.

Müəllimlər - müəllimlər şagirdlərin təlim, tərbiyə və təhsilində baş rolda olan, birbaşa cavabdeh olan şəxslərdir. Onlar siniflərdə öyrətmə fəaliyyətlərini icra edib eyni zamanda öyrənilməyə nəzarət edirlər.

Məktəbin idarəedilməsində müəyyən qanunlar əsasında həyata keçirilməsində təhsil sektorunun xüsusiyyətləri nəzərə alınır. Məktəb rəhbərliyinin fəaliyyətini digər müəssisələrinin rəhbərlərinin işindən xeyli fərqliliyi ilə nəticələndirmək. Biz burada “Təhsil sisteminin idarə olunmasının nəzəriyyəsi və metodologiyası” fəslində məktəblərin idarə edilməsində əsas məqsəd kimi ümumi idarəetmə nəzəriyyəsinə baxacağıq.

Aydın məsələdir ki, məktəb idarəçiliyinin təkmilləşdirilməsi üçün onun strukturu, idarəetmə işinin optimallaşdırılması prosesinin elmi əsaslarla davamlı təkmilləşdirilməsi ilə nail mümkündür.

Məktəbin konkret şəraitində məharətlə idarə edilməsi və məktəb rəhbərliyinin tədris prosesinin gedişinə məqsədyönlü nəzarətinin təmin edilməsi, müəyyən düzəlişlərin verilməsi ümumi idarəetmə nəzəriyyəsinin ən mühüm müddəalarından biri hesab etmək olar.

Məktəbdə idarəetmə prosesinin xüsusiyyətlərindən biri də ondan ibarətdir ki, direktor və direktor müavinləri özləri birbaşa olaraq pedaqoji fəaliyyətin iştirakçıları olaraq, subyektləri olan iki obyektin (müəllim, şagird) fəaliyyətinə rəhbərlik edir.

Məktəb direktoru kollektivin rəhbəri olmaqla yanaşı o, həm də kollektivin bir üzvüdür, yəni o da müəllimdir. Məhz bundan ötrü o, elmi-pedaqoji və metodiki cəhətdən, təlim-tərbiyə işlərinə yeniliklər gətirilməsində, nizam-intizamın saxlanılmasında şagirdlər və valideynlərlə ünsiyyət zamanı münasibətdə hər bir işçiyə nümunə olmalıdır.

O, pedaqoji kollektivin rəhbəri işçiləri ilə tez-tez əlaqə saxlamalı, onlarla söhbətlər aparmalı, onların arzu-istəklərinə və iradlarını dinləməli, o cümlədən faydalı təkliflərinə diqqətlə yanaşmalı və lazım gəldikdə həmin təklifləri qiymətləndirməlidir. Əks təqdirdə pedaqoji kollektivlə rəhbərlik arasında müəyyən uyuşmazlıq meydana çıxar və qarşıya qoyulmuş məqsədə çatmaq bir o qədər çətin olar.

Təəssüf ki, bəzən kollektivlərdə rəhbərlə (direktor, direktor müavinləri və s.) işçilər arasında münasibətlər o qədər də ürəkəçan olmur hansı ki, bu da ümumi işə mənfi təsir göstərir və nəticədə idarəetmədə keyfiyyətin aşağı düşməsi ilə nəticələnir.

Rəhbər işçinin yol verdiyi ciddi nöqsanlara özbaşınalıq, tənqidi boğmaq, əməkdaşlara etibar etməmək, kobud ünsiyyət, yaltaqlara meydan vermək, əmrilər vermək, yenilikləri qəbul etməkdən imtina etmək və s. aiddir. Məktəb direktoru özünə nəzarəti, öz işinə həssaslıqla yanaşması, hətta şəxsi keyfiyyətlərinə dair ünvanlanan tənqidi münasibətləri qəbul etməsi indiki şəraitdə daha vacibdir. Çünki özünə olan tənqidləri qəbul etməyən heç bir rəhbər işçi bir şəxsiyyət kimi gözdən tez düşür. Rəhbər işçi kollektiv arasında nəinki mehribanlıq təşkil etməlidir, həm də ona nail olmalıdır ki, kollektiv öz-özünü təşkil etsin. Məktəbə rəhbərlik etmək olduqca mürəkkəb və qəliz işdir.

Məktəb direktorunun vəzifə öhdəliklərinə aiddir: idarəetmə fəaliyyətində təlim-tərbiyə işlərinin rəhbərliyi, metodik işin rəhbərliyi, pedaqoji kollektiv ilə birgə fəaliyyətinə rəhbərliyi. Həmçinin məktəb rəhbərliyinin (direktorunun) idarəçilik fəaliyyətinin ən mühüm

istişaməti kollektivin bütün müəllimləri, şagird və valideyn kollektivlərini idarəetməyə cəlb etmə bacarığı, məktəb fəaliyyətinin əsas məqsədinə çatmaq üçün onların təklif, iradları və göstərdikləri səyləri kollektiv yaradıcı fəaliyyətdə birlikdə komanda işi təşkil etməkdən ibarətdir.

Məktəbin direktoru bərabər fəaliyyət göstərdiyi müəllim kollektivi ilə birlikdə fəaliyyətin əsas məqsədləri ilə yaxından tanış olmalı və mənimsəməlidir. Məktəb kollektivinin fəaliyyətini məktəbin konkret şəraiti nəzərə alınmaqla təşkil etmək, pedaqoji kollektivlə ardıcıl iş sistemini təşkil etmək, məktəbdaxili informasiya sistemini yaratmaq, məktəbin qarşılıqlı fəaliyyətini təşkil etmək, Şagirdlərin bir şəxsiyyət kimi formalaşması, inkişafı üçün şərait, onların şəxsi müqəddəratını təyin etmək məqsəduyğundur.

İdarəetmə dedikdə aşağıdakı fəaliyyətlər nəzərdə tutulur :

1. İdarəetmənin quruluşunun müəyyənləşdirilməsi;
2. Hər dövr üçün quruluş, məzmun və funksional komponentlərin ayrılması;
3. İdarəetmənin bütün dövrlərində işdə olan çətinlikləri aradan qaldırmaq üçün fəaliyyətin təşkil edilməsi;
4. Pedaqoji və şagird kollektivləri səviyyəsində idarəetmə bacarığının və şagird kollektivləri səviyyəsində idarəetmə bacarığının öyrədilib həyata keçirilməsi;

II FƏSİL. Verilənlər Bazasının qurulması

2.1. Verilənlər Bazasının struktur elementləri

Verilənlər bazası kompüter sistemində təşkil edilən və saxlanılan məlumatların strukturlaşdırılmış toplusudur. Verilənlər bazası məlumatları səmərəli şəkildə idarə etmək, saxlamaq, əldə etmək və yeniləmək üçün nəzərdə tutulmuşdur. O, adətən kompüter sistemində saxlanılır və verilənlər bazası idarəetmə proqramından istifadə etməklə idarə oluna bilər. Maliyyə əməliyyatlarının və müştəri məlumatlarının idarə edilməsindən tutmuş elmi tədqiqat məlumatlarının saxlanmasına qədər geniş məqsədlər üçün istifadə edilə bilər. Verilənlər bazası müxtəlif növ ola bilər, məsələn, əlaqəli verilənlər bazası, NoSQL verilənlər bazası, qrafik verilənlər bazası, SQL verilənlər bazası və s.

Verilənlər bazası adətən bir və ya bir neçə cədvəldən ibarətdir ki, onların hər birində konkret mövzu və ya mövzu üzrə məlumatlar var. Hər bir cədvəl sətirlərdən (həmçinin qeydlər adlanır) və sütunlardan (həmçinin sahələr adlanır) ibarətdir. Sətirlər saxlanılan məlumatların fərdi nümunələrini, sütunlar isə həmin məlumatların müxtəlif atributlarını və ya xüsusiyyətlərini təmsil edir.

Model-View-Controller (MVC) nümunəsi istifadəçi interfeysləri ilə tətbiqlərin layihələndirilməsi və inkişaf etdirilməsində geniş istifadə olunan proqram memarlıq nümunəsidir. O, proqramı bir-biri ilə əlaqəli üç komponentə ayırır: model, görünüş və nəzarətçi. Hər bir komponentin fərqli məsuliyyətləri var və digərləri ilə əlaqələndirilmiş şəkildə qarşılıqlı əlaqədə olur.

MVC nümunəsindəki hər bir komponentin qısa icmalı buradadır.

Model: Model verilənləri və tətbiqin biznes məntiqini təmsil edir. O, tətbiqin məlumatlarını əhatə edir və ona daxil olmaq və manipulyasiya etmək üçün üsullar təqdim edir. O, istifadəçi interfeysindən və təqdimat məntiqindən müstəqildir. Sadəcə dillə desək, model tətbiqin domenə xas bilik və qaydalarını təmsil edir.

Görünüş: Görünüş istifadəçi interfeysinin göstərilməsi və məlumatların istifadəçiyə təqdim edilməsi üçün məsuliyyət daşıyır. Modeldən məlumat alır və onu istifadəçi üçün uyğun formatda göstərir. Görünüş passivdir və heç bir biznes məntiqi yoxdur. O, sadəcə olaraq model tərəfindən verilən məlumatları təqdim edir və istifadəçi hərəkətlərini nəzarətçiyə göndərir.

Nəzarətçi: Nəzarətçi model və görünüş arasında vasitəçi kimi çıxış edir. Görünüşdən istifadəçi hərəkətlərini qəbul edir, model üzərində lazımi əməliyyatları yerinə yetirir və buna uyğun olaraq görünüşü yeniləyir. Nəzarətçi istifadəçi daxiletmələrinin idarə edilməsi, qərarların qəbul edilməsi və model və görünüş arasında məlumat axınının əlaqələndirilməsi üçün tətbiqin məntiqini ehtiva edir.

MVC modelinin əsas ideyası narahatlıqları ayırmaq və komponentlər arasında boş birləşməni təşviq etməkdir. Bu ayırma tətbiqin daha asan saxlanmasına, sınaqdan keçirilməsinə və genişlənməsinə imkan verir. Görünüş və ya model kimi bir komponentdəki dəyişikliklər, komponentlər arasındakı interfeyslər dəyişməz qaldıqda, digərlərinə birbaşa təsir göstərmir.

Ümumiyyətlə, MVC nümunəsi proqramları, xüsusən də istifadəçi interfeysi olanları modul və dəyişdirilə bilən komponentlərə bölmək yolu ilə təşkil etmək və dizayn etmək üçün strukturlaşdırılmış bir yol təqdim edir.

MODELS

Bu, Django veb çərçivəsi ilə yazılmış Python kodudur. O, dörd modeli müəyyən edir: Kateqoriya, Müəllim, Problem və Problem Filtri.

Model məlumatlarınız haqqında yeganə, qəti məlumat mənbəyidir. O, saxladığınız məlumatların əsas sahələrini və davranışlarını ehtiva edir. Hər bir model vahid verilənlər bazası cədvəlinə uyğunlaşdırılır.

Kateqoriya: Problemlər üçün kateqoriyanı təmsil edir. Onun kateqoriya adını saxlamaq üçün ad sahəsi (CharField) və kateqoriya haqqında əlavə məlumat saxlamaq üçün məlumat sahəsi (TextField) var.

Müəllim: Müəllimi təmsil edir. O, müəllimin adını saxlamaq üçün ad sahəsinə (CharField), tədris etdiyi fənni saxlamaq üçün fənn sahəsinə (CharField) və müəllim haqqında əlavə məlumatı saxlamaq üçün məlumat sahəsinə (TextField) malikdir.

Problem: Problemi təmsil edir. Problemin adını saxlamaq üçün ad sahəsi (CharField), problemi onu yaradan müəllimlə əlaqələndirmək üçün yaradılmış_by sahəsi (ForeignKey), problemin həll olunub-olunmadığını göstərmək üçün həll edilmiş sahə (BooleanField), yaradılmış_at sahəsi var. Problemin yaranma tarixini və vaxtını saxlamaq üçün (DateTimeField), problemin həlli üçün görülən tədbirləri saxlamaq üçün alınan tədbirlər

sahəsi (TextField), problemi kateqoriya ilə əlaqələndirmək üçün kateqoriya sahəsi (ForeignKey) və nəticə sahəsi (TextField)) problemin həllinin nəticəsini saxlamaq. O, həmçinin problemin adının sətir təsvirini qaytarmaq üçün __str__ metodunu və problemin təfərrüatlı görünüşü üçün URL əldə etmək üçün get_absolute_url metodunu müəyyən edir.

ProblemFilter: Problemlər üçün filtri təmsil edir. Onun filtr adını saxlamaq üçün ad sahəsi (CharField), başlanğıc_tarix və bitmə_tarixi sahələri (DateTimeField) yaranma tarixlərinə əsaslanan problemləri süzgəcdən keçirmək üçün, kateqoriyaya əsasən problemləri filtrləmək üçün kateqoriya sahəsi (ForeignKey), müəllim sahəsi (ForeignKey) var. problemləri onları yaradan müəllim əsasında filtrləyin və problemlərin həll olunub-olunmamasına görə süzgəcdən keçirmək üçün həll edilmiş sahə (BooleanField). O, həmçinin filtr meyarlarına əsasən süzülmüş problemləri əldə etmək üçün get_problems metodunu müəyyənləşdirir.

Modellər CharField, TextField, BooleanField və DateTimeField kimi Djanqonun modellər modulu tərəfindən təmin edilən müxtəlif sahə növlərindən istifadə edir. Onlar həmçinin Müəllim, Kateqoriya və Problem modelləri arasında əlaqə yaratmaq üçün ForeignKey-dən istifadə edən modellər arasındakı əlaqələrdən istifadə edirlər.

django.urls-dən əks funksiya problem modelinin get_absolute_url metodunda onun ID əsasında problemin təfərrüatlı görünüşü üçün URL yaratmaq üçün istifadə olunur.

Ümumiyyətlə, bu modellər Django proqramında məlumatlarınızın strukturunu və əlaqələrini müəyyən edir.

ProblemFilter modeli filtrləmə meyarları dəsti ilə get_problems() metodunu çağırmaqla Problem obyektlərinin süzülmüş sorğu dəstini qaytara bilir.

ProblemFilter modelindən istifadə nümunəsi:

Bəzi filtrləmə meyarları ilə problemləli filtr nümunəsi yaradaq

```
problem_filtr = ProblemFilter(ad="Problem 1", həll olundu=Doğru)
```

Süzgəcdən keçirilmiş problemlər toplusunu əldə edək

```
filtered_problems = problem_filtr.get_problems()
```

Süzgəcdən keçirilmiş problemlər sorğu dəsti üzərində təkrarlacaq

filtered_problems-də problem üçün: çap (problem.adı, problem.adla yaradılmış_ad, problem.kateqoriya.adı, problem.həll edildi).

VIEWS

ADD CATEGORY

Django şablon dilində yazılmış kod parçasını təqdim etmişik. Bu kod 'base.html' adlı əsas şablonu genişləndirir və onun daxilində 'content' adlı bloku müəyyən edir.

"Məzmun" blokunda kateqoriya əlavə etmək üçün HTML forması var. Kodu parçalayaq: Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

Bu sətirdə "Kateqoriyaya əlavə et" mətnini əks etdirən HTML başlıq elementi (<h1>) var.

```
<h1>Kateqoriya əlavə etmək</h1>
```

Bu kod parçası HTML formasını təmsil edir. Metod atributu forma məlumatlarının HTTP POST sorğusu vasitəsilə göndəriləcəyini göstərən "yazıya" təyin edilib. {% csrf_token %} şablon teqi təhlükəsizlik məqsədləri üçün formada CSRF (Saytlarası Sorğu Saxtakarlığı) tokeninin daxil edilməsini təmin edir.

Formaya form.name adı ilə sahə üçün etiket daxildir. {{ form.name }} dəyişəni kateqoriya adını daxil etmək üçün giriş sahəsini təqdim edir. type="submit" atributu "Əlavə et" etiketi ilə göndərmə düyməsi yaradır.

```
<form method="post">
  {% csrf_token %}
  <label for="{{ form.name.auto_id }}">Kateqirynın adı :</label>
  {{ form.name }}
  <input type="submit" value="Добавить">
</form>
```

Bu sətir "məzmun" blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və kateqoriya əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir və istifadəçilərə kateqoriya adını daxil edib onu təqdim etməyə imkan verir.

```
{% endblock %}
```

Bu sətir "məzmun" blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və kateqoriya əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir və istifadəçilərə kateqoriya adını daxil edib onu təqdim etməyə imkan verir.

ADD PROBLEM

```

1  {% extends "base.html" %}
2
3  {% block content %}
4  <div class="container">
5      <h1>Problem əlavə etmək</h1>
6      <form method="POST" action="{% url 'add_problem' %}">
7          {% csrf_token %}
8          {{ form.as_p }}
9          <button type="submit" class="btn btn-primary">Təsdiqlə</button>
10     </form>
11     {% if form.errors %}
12     <div class="alert alert-danger" role="alert">
13         {{ form.errors }}
14     </div>
15     {% endif %}
16 </div>
17 {% endblock %}

```

Şəkil 2.1 " Problemin əlavə edilmə" bloku

Təqdim olunan kod parçası Django şablon dilində yazılmışdır. O, "base.html" adlı əsas şablonu genişləndirir və "content" adlı bloku müəyyənləşdirir.

Kodu parçalayaq:

```
{% extends "base.html" %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq "base.html" şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```

<div class="container">
  <h1>Problem əlavə etmək</h1>
  <form method="POST" action="{% url 'add_problem' %}">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" class="btn btn-primary">Təsdiqlə</button>
  </form>
  {% if form.errors %}
  <div class="alert alert-danger" role="alert">
    {{ form.errors }}
  </div>
  {% endif %}
</div>
z

```

Şəkil 2.2 " Problemin əlavə edilmə 2ci hissə" bloku

"Məzmun" blokunda problem əlavə etmək üçün HTML forması var. Budur onun elementlərinin bölgüsü:

<div class="container">: Bu div elementi üslub məqsədləri üçün "konteyner" sinfinə malikdir.

<h1>Problemin əlavə edilməsi</h1>: Bu sətirdə "Problem əlavə etmək" başlığı göstərilir.

<form metodu="POST" action="{% url 'add_problem' %}">: Bu sətir HTTP metodu "POST" olaraq təyin edilmiş HTML formasını müəyyən edir. Formanın fəaliyyət atributu forma məlumatlarının təqdim ediləcəyi URL-i müəyyən edir. {% url 'add_problem' %} şablon teqi "add_problem" adlı URL nümunəsi əsasında dinamik olaraq URL yaratmaq üçün istifadə olunur.

{% csrf_token %}: Bu şablon teqi təhlükəsizlik məqsədləri üçün formada CSRF (Saytlararası Sorğu Saxtakarlığı) tokeninin daxil edilməsini təmin edir.

{{ form.as_p }}: Bu dəyişən forma sahələrini <label> teqlərinə bükülmüş HTML paraqrafları (<p>) kimi təqdim edərək formanın sadə təsvirini yaradır.

<button type="submit" class="btn btn-primary">Təsdiqlə</button>: Bu sətir "Təsdiqlə" etiketli və "btn" CSS sinifləri olan təqdim düyməsini təmsil edir. üslub məqsədləri üçün "btn-primary".

{% if form.errors %}: Bu sətir hər hansı forma xətalalarının olub olmadığını yoxlayır.

<div class="alert alert-danger" role="alert">: Forma xətalı varsa, "alert" və "alert-danger" sinifləri olan bu div elementi göstərilir.

`{{ form.errors }}`: Bu dəyişən div elementində forma xəalarını göstərir.

```
{% endblock %}
```

Bu xətt "məzmun" blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və problem əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir. Forma abzaslara bükülmüş sahələri ilə göstərilir və təqdim düyməsi verilir. Forma xəaları varsa, onlar xəbərdarlıq qutusunda göstərilir.

ADD TEACHER

Təqdim olunan kod parçası Django şablon dilində yazılmışdır. O, 'base.html' adlı əsas şablonu genişləndirir və 'content' adlı bloku müəyyən edir. Kodu parçalayaq:

```
{% extends 'base.html' %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq 'base.html' şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```
<h1>Müəllim əlavə et</h1>
```

Bu sətirdə "Müəllim əlavə et" mətnini əks etdirən HTML başlıq elementi (<h1>) var .

```
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Əlavə et">
</form>
```

Bu kod parçası müəllim əlavə etmək üçün HTML formasını təmsil edir. Metod atributu forma məlumatlarının HTTP POST sorğusu vasitəsilə göndəriləcəyini göstərən "yazıya" təyin edilib. `{% csrf_token %}` şablon teqi təhlükəsizlik məqsədləri üçün formada CSRF (Saytlarası Sorğu Saxtakarlığı) tokeninin daxil edilməsini təmin edir.

Formaya `{{ form.as_p }}` tərəfindən yaradılan sahələr daxildir. Bu, HTML paraqraflarına (`<p>`) bükülmüş forma sahələrini etiketləri ilə göstərəcək.

Sonuncu sətirdə "göndər" tipli giriş elementi var. Kliklədikdə, bu giriş formanı təqdim edəcək.

```
{% endblock %}
```

Bu sətir "məzmun" blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və müəllim əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir. Forma abzaslara bükülmüş sahələri ilə göstərilir və təqdim düyməsi verilir.

EDIT PROBLEM

Təqdim olunan kod parçası Django şablon dilində yazılmışdır. O, "base.html" adlı əsas şablonu genişləndirir və "content" adlı bloku müəyyənləşdirir.

Kodu parçalayaq:

```
{% extends "base.html" %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq "base.html" şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir “content” adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```
<div>
  <h1>Redakte et</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Dəyişikləri tədiq et</button>
  </form>
</div>
```

Şəkil 2.3 " Redaktə et" bloku

"Məzmun" blokunda məzmunu redaktə etmək üçün HTML forması var. Budur onun elementlərinin bölgüsü:

`<div>`: Bu div elementi konteyner kimi istifadə olunur.

`<h1>Redakte et</h1>`: Bu sətirdə "Redakte et" başlığı göstərilir.

`<form metodu="post">`: Bu sətir HTTP metodu "post" olaraq təyin edilmiş HTML formasını müəyyən edir. Forma POST sorğusundan istifadə edərək məlumatları təqdim edəcək.

`{% csrf_token %}`: Bu şablon teqi təhlükəsizlik məqsədləri üçün formada CSRF (Saytlararası Sorğu Saxtakarlığı) tokeninin daxil edilməsini təmin edir.

`{{ form.as_p }}`: Bu dəyişən forma sahələrini `<label>` teqlərinə bükülmüş HTML paraqrafları (`<p>`) kimi təqdim edərək formanın sadə təsvirini yaradır.

`<button type="submit">Dəyişikləri tədiq et</button>`: Bu sətir "Dəyişikləri tədiq et" etiketli təqdim düyməsini təmsil edir .Bu düymə kliklədikdə formanı təqdim edəcək.

```
{% endblock %}
```

Bu xətt "məzmun" blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və məzmunu redaktə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir. Forma abzaslara bükülmüş sahələri ilə göstərilir və təqdim düyməsi verilir.

FİLTƏR PROBLMƏS

Təqdim etdiyimiz kod Python veb çərçivəsi olan Django-da yazılmış şablon kimi görünür. O, "base.html" adlı əsas şablonu genişləndirir və genişləndirilmiş şablon tərəfindən doldurulacaq "məzmun" adlı bloku müəyyən edir. Şablon kodunun bölgüsü budur:

```
{% extends "base.html" %}
```

Bu sətir müəyyən edir ki, bu şablon "base.html" şablonunu genişləndirir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bu blokun içindəki məzmun "base.html" şablonunda müvafiq bloka daxil ediləcək.

```
<div>
  <h1>Filter Problems</h1>
  <form method="POST" action="{% url 'filter_problems' %}">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Filtr</button>
  </form>
  {% if page_obj %}
  <h2>Problem filtri</h2>
  <ul>
    {% for problem in page_obj %}
    <li>
      <a href="{% url 'view_problem' problem.id %}">{{ problem.name }}</a>
      <p>{{ problem.measures_taken }}</p>
      <p>Category: {{ problem.category.name }}</p>
      <p>Teacher: {{ problem.created_by.name }}</p>
    </li>
    {% endfor %}
  </ul>
  {% endif %}
</div>
```

Şəkil 2.4 " Filterləmə" bloku

"Məzmun" blokunun içərisində HTML kodunuz və Django şablon teqləri var. Gəlin onu parçalayaq:

<div> elementi bu şablonun məzmununu ehtiva edir.

<h1>Filtr Problemləri</h1> formanın başlığıdır.

<form metodu="POST" action="{% url 'filter_problems' %}"> "filter_problems" URL-inə POST metodundan istifadə etməklə təqdim olunacaq formanı işə salır. {% url 'filter_problems' %} şablon teqi göstərilən görünüş adına əsaslanaraq URL yaradır.

{% csrf_token %} formaya Saytlararası Sorğu Saxtakarlığı (CSRF) nişanı daxil edən şablon teqdir. Bu, icazəsiz sorğuların qarşısını almaq üçün təhlükəsizlik tədbiridir.

{{ form.as_p }} forma sahələrini etiketlərlə HTML paraqrafları (<p>) kimi göstərir. Forma dəyişəninin şablon kontekstinə ötürüldüyü fərz edilir.

<button type="submit">Filtr</button> forma üçün göndərmə düyməsidir.

{% if page_obj %} "page_obj" dəyişəninin mövcud olub olmadığını yoxlayır.

<h2>Problem filtri</h2> süzülmüş problemlərin siyahısı üçün başlıqdır.

 sıralanmamış siyahıya başlayır.

{% for problem in page_obj %} "page_obj" dəyişəninəki hər "problem" üzərində dövrə başlayır.

 siyahı elementinə başlayır.

{{ problem.name }} problem təfərrüatları səhifəsinə baxmaq üçün keçiddir. URL "view_problem" görünüş adından istifadə etməklə və problemin ID-sini parametr kimi ötürməklə yaradılır.

<p>{{ problem.measures_taken }}</p> problem üçün görülən tədbirləri göstərir.

<p>Kateqoriya: {{ problem.category.name }}</p> problemin kateqoriya adını göstərir.

<p>Müəllim: {{ problem.created_by.name }}</p> problemi yaradan müəllimin adını göstərir.

{% endfor %} problemlər üzərində dövrəni bitirir.

 sıralanmamış siyahını bitir.

{% endif %} if ifadəsini "page_obj" varlığını yoxlayır.

Ümumiyyətlə, bu şablon problemlərin filtrasiyası üçün forma verir.

İNDEKS

Təqdim etdiyimiz kod "base.html" adlı əsas şablonu genişləndirən Django şablonu kimi görünür. O, bu şablonun məzmununun əsas şablona daxil ediləcəyi "məzmun" adlı bloku müəyyən edir. Şablon kodunun bölgüsü budur:

```
{% extends 'base.html' %}
```

Bu sətir müəyyən edir ki, bu şablon "base.html" şablonunu genişləndirir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bu blokun içindəki məzmun "base.html" şablonunda müvafiq bloka daxil ediləcək.

```
<h1 class='main-title'>Əsas səhifə</h1>
<p class='main-stat'>Problem sayı: {{ problem_count }}</p>
<p class='main-stat'>Kateqoriya sayı: {{ category_count }}</p>
<p class='main-stat'>Müəllim sayı: {{ teacher_count }}</p>

<h2 class='main-subtitle'>Yeni problemlər</h2>

{% for problem in problems %}
<div class='problem-card'>
  <div class='problem-info'>
    <h5 class='problem-name'>{{ problem.name }}</h5>
    <p class='problem-detail'>Kim yaradıb: {{ problem.created_by }}</p>
    <p class='problem-detail'>Həll olunub?: {{ problem.solved }}</p>
    <p class='problem-detail'>Nə vaxt yaranıb: {{ problem.created_at }}</p>
    <p class='problem-detail'>Nə tədbir görüldü: {{ problem.measures_taken }}</p>
    <p class='problem-detail'>Kateqoriya: {{ problem.category }}</p>
    <p class='problem-detail'>Nəticə: {{ problem.result }}</p>
  </div>
  <a href="{% url 'edit_problem' problem.id %}" class='problem-edit'>Redakte et</a>
  <form method='POST' action="{% url 'delete_problem' problem.id %}" class='problem-delete'>
    {% csrf_token %}
    <button type='submit' class='problem-delete-button'>Problemi sil
  </button>
  </form>
</div>
{% endfor %}
```

Şəkil 2.5 "Filterləmə" bloku

"Məzmun" blokunun içərisində HTML kodunuz və Django şablon teqləri var. Gəlin onu parçalayaq:

```
<h1 class='main-title'>Əsas səhifə</h1> əsas səhifənin başlığıdır.
```

`<p class='main-stat'>Problem sayı: {{ problem_count }}</p>` "problem_count" dəyişəninin dəyərindən istifadə edərək problemlərin sayını göstərir.

`<p class='main-stat'>Kateqoriya sayı: {{category_count }}</p>` "category_count" dəyişəninin dəyərindən istifadə edərək kateqoriyaların sayını göstərir.

`<p class='main-stat'>Müəllim sayı: {{ teacher_count }}</p>` "teacher_count" dəyişəninin dəyərindən istifadə edən müəllimlərin sayını göstərir.

`<h2 class='main-subtitle'>Yeni inkişaf</h2>` yeni problemlər bölməsinin başlığıdır.

`{% for problem in problem %}` "problemlər" dəyişənindəki hər bir "problem" üzərində dövrə başlayır.

`<div class='problem-card'>` problem üçün konteyneri işə salır.

`<div class='problem-info'>` problemin məlumatını ehtiva edir.

`<h5 class='problem-name'>{{ problem.name }}</h5>` problemin adını göstərir.

`<p class='problem-detail'>Kim yaradıb: {{ problem.created`

Təqdim etdiyimiz kod "base.html" adlı əsas şablonu genişləndirən Django şablonu kimi görünür. O, bu şablonun məzmununun əsas şablona daxil ediləcəyi "məzmun" adlı bloku müəyyən edir. Şablon kodunun bölgüsü budur:

```
{% extends 'base.html' %}
```

Bu sətir müəyyən edir ki, bu şablon "base.html" şablonunu genişləndirir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bu blokun içindəki məzmun "base.html" şablonunda müvafiq bloka daxil ediləcək.

```

<h1>Kateqoriyaların siyahısı</h1>
<a href="{% url 'add_category' %}">Kateqoriya əlavə et</a>
<table>
  <thead>
    <tr>
      <th>Adı</th>
      <th>Problemlərin sayı</th>
      <th>Həll olunan problem sayı</th>
      <th>İşləmlər</th>
    </tr>
  </thead>
  <tbody>
    {% for category in categories %}
    <tr>
      <td>{{ category.name }}</td>
      <td>{{ category.problem_count }}</td>
      <td>{{ category.solved_problem_count }}</td>
      <td><a href="{% url 'view_category' category.id %}">Baxmaq</a></td>
    </tr>
    {% endfor %}
  </tbody>
</table>
{% endblock %}

```

Şəkil 2.6 "Kateqoriya əlavə et" bloku

"Məzmun" blokunun içərisində HTML kodunuz və Django şablon teqləri var. Gəlin onu parçalayaq:

`<h1>Kateqoriyaların siyahısı</h1>` kateqoriyalar siyahısı üçün başlıqdır.

`Kateqoriya əlavə et` yeni kateqoriya əlavə etmək üçün keçiddir. URL "add_category" görünüş adı ilə yaradılır.

`<table>` kateqoriyaları göstərmək üçün cədvələ başlayır.

`<thead>` cədvəlin başlıq bölməsini başlayır.

`<tr>` cədvəl sırasına başlayır.

`<th>Adı</th>` kateqoriya adı üçün cədvəl başlıq xanasını təyin edir.

`<th>Problemlərin sayı</th>` problemlərin sayı üçün cədvəl başlığı xanasını təyin edir.

`<th>Həll olunan problem sayı</th>` həll edilmiş problemlərin sayı üçün cədvəl başlığı xanasını təyin edir.

`<th>İşləmlər</th>` əməliyyatlar üçün cədvəl başlıq xanasını təyin edir.

`</tr>` cədvəl sırasını bitirir.

`</thead>` cədvəlin başlıq bölməsini bitirir.

`<tbody>` cədvəlin əsas hissəsini başlayır.

`{% kateqoriyalar üçün kateqoriya % }` "kateqoriyalar" dəyişənindəki hər bir "kateqoriya" üzərində dövrə başlayır.

`<tr>` cədvəl sırasına başlayır.

`<td>{{ kateqoriya.adi }}</td>` cədvəl xanasında kateqoriyanın adını göstərir.

`<td>{{ kateqoriya.problem_count }}</td>` cədvəl xanasındakı problemlərin sayını göstərir.

`<td>{{ category.solved_problem_count }}</td>` cədvəl xanasında həll edilmiş problemlərin sayını göstərir.

`<td>Baxmaq</td>` kateqoriya təfərrüatlarına baxmaq üçün keçid göstərir. URL "view_category" görünüş adından istifadə etməklə və kateqoriyanın ID-sini parametr kimi ötürməklə yaradılır.

`</tr>` cədvəl sırasını bitirir.

`{% endfor % }` kateqoriyalar üzərində dövrəni bitir.

`</tbody>` cədvəlin əsas hissəsini bitirir.

LİST CATEGORIES

Təqdim olunan kod parçası 'base.html' adlı əsas şablonu genişləndirən Django şablonudur. O, problem statistikasını, yeni problemlərin siyahısını və kateqoriyaların siyahısını daxil etməklə, əsas səhifə məzmununu göstərmək üçün HTML işarələməsini ehtiva edən "məzmun" adlı bloku müəyyən edir.

Kodu parçalayaq:

```
{% extends 'base.html' %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq 'base.html' şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```

<h1 class='main-title'>Əsas səhifə</h1>
<p class='main-stat'>Problem sayı: {{ problem_count }}</p>
<p class='main-stat'>Kateqoriya sayı: {{ category_count }}</p>
<p class='main-stat'>Müəllim sayı: {{ teacher_count }}</p>

```

Bu sətirlərdə problemlər, kateqoriyalar və müəllimlər haqqında əsas başlıq və bəzi statistik məlumatlar göstərilir. Statistika şablon dəyişənlərindən ({{ problem_count }}, {{ category_count }}, {{ teacher_count }}) istifadə edilməklə təqdim edilir ki, onların şablon göstərilərkən təqdim edilməsi gözlənilir.

```

<h2 class='main-subtitle'>Yeni problemlər</h2>

{% for problem in problems %}
<div class='problem-card'>
  <div class='problem-info'>
    <h5 class='problem-name'>{{ problem.name }}</h5>
    <p class='problem-detail'>Kim yaradıb: {{ problem.created_by }}</p>
    <p class='problem-detail'>Həll olunub?: {{ problem.solved }}</p>
    <p class='problem-detail'>Nə vaxt yaranıb: {{ problem.created_at }}</p>
    <p class='problem-detail'>Nə tədbir görüldü: {{ problem.measures_taken }}</p>
    <p class='problem-detail'>Kateqoriya: {{ problem.category }}</p>
    <p class='problem-detail'>Nəticə: {{ problem.result }}</p>
  </div>
  <a href="{% url 'edit_problem' problem.id %}" class='problem-edit'>Redakte et</a>
  <form method='POST' action="{% url 'delete_problem' problem.id %}" class='problem-delete'>
    {% csrf_token %}
    <button type='submit' class="problem-delete-button">Problemi sil
  </button>
  </form>
</div> {% endfor %}

```

Şəkil 2.7 "Problem əlavə et" bloku

Bu sətirlərdə “Yeni problem ” adlı bölmə göstərilir və problemlərin siyahısı üzərində təkrarlanır. Hər bir problem üçün o, 'problem-card' sinfi ilə <div> elementi yaradır və şablon dəyişənlərindən ({{ problem.name }}, {{ problem.created_by }} və s.) istifadə edərək problem haqqında müxtəlif təfərrüatları göstərir. Buraya həmçinin problemi redaktə etmək üçün keçidlər və problemi silmək üçün forma daxildir.

Bu sətirlərdə "Kateqiyalar(problem sayı)" başlıqlı bölmə göstərilir və siyahı üzərində təkrarlanır.

LİST PROBLEMS

Təqdim olunan kod parçası 'base.html' adlı əsas şablonu genişləndirən Django şablonudur. O, problem statistikasını, yeni problemlərin siyahısını və kateqoriyaların siyahısını daxil etməklə, əsas səhifə məzmununu göstərmək üçün HTML işarələməsini ehtiva edən "məzmun" adlı bloku müəyyən edir.

Kodu parçalayaraq:

```
{% extends 'base.html' %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq 'base.html' şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.


```

<h1>Kateqoriyaların siyahısı</h1>
<a href="{% url 'add_category' %}">Kateqoriya əlavə et</a>
<table>
  <thead>
    <tr>
      <th>Adı</th>
      <th>Problemlərin sayı</th>
      <th>Həll olunan problem sayı</th>
      <th>İşləmlər</th>
    </tr>
  </thead>
  <tbody>
    {% for category in categories %}
    <tr>
      <td>{{ category.name }}</td>
      <td>{{ category.problem_count }}</td>
      <td>{{ category.solved_problem_count }}</td>
      <td><a href="{% url 'view_category' category.id %}">Baxmaq</a></td>
    </tr>
    {% endfor %}
  </tbody>
</table>

```

Şəkil 2.8 " Kateqoriyaların siyahısı " bloku

Bu sətirlərdə "Kateqoriyaların siyahısı" başlığı ilə bölmə göstərilir. O, həmçinin kateqoriya əlavə etmək üçün bir keçid daxildir.

Kateqoriya siyahısının strukturu üçün <table> elementindən istifadə olunur. Onun cədvəl başlıqları üçün <thead> bölməsi və cədvəl sıraları üçün <tbody> bölməsi var.

Cədvəl başlıqları (<th>) aşağıdakı kimi müəyyən edilir:

"Adı": Bu kateqoriyanın adını təmsil edir.

"Problemlərin sayı": Bu kateqoriya ilə əlaqəli problemlərin sayını göstərir.

"Həll olunan problem sayı": Bu kateqoriyada həll edilmiş problemlərin sayını göstərir.

"İşləmlər": Bu kateqoriya üzrə yerinə yetirilə bilən hərəkətləri təmsil edir.

<tbody> bölməsində dövrə kateqoriyalar dəyişəni üzərində təkrarlanır və hər bir kateqoriya üçün cədvəl cərgəsi (<tr>) yaradır. Cədvəl verilənləri (<td>) elementləri kateqoriya adını, problem sayını, həll edilmiş problem sayını və kateqoriya təfərrüatlarına baxmaq üçün keçidi göstərir.

Kateqoriya təfərrüatlarına baxmaq üçün link parametr kimi kateqoriya ID ilə 'view_category' adlı URL nümunəsindən istifadə etməklə yaradılır.

```
{% endblock %}
```

Bu sətir "məzmun" blokunu bitirir.

Ümumiyyətlə, şablon kateqoriyaların siyahısını müvafiq məlumatlarla cədvəl formatında göstərir və yeni kateqoriya əlavə etmək üçün keçid təqdim edir.

LİST TEACHERS

Təqdim olunan kod parçası 'base.html' adlı əsas şablonu genişləndirən Django şablonudur. O, bütün problemlərin cədvəlini göstərmək üçün HTML işarələməsini ehtiva edən "məzmun" adlı bloku müəyyən edir.

Kodu parçalayaq:

```
{% extends 'base.html' %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq 'base.html' şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```

<h1>Bütün problemlər</h1>

<p><a href="{% url 'add_problem' %}">Добавить проблему</a></p>

<table>
  <tr>
    <th>Adı</th>
    <th>Kateqoriya</th>
    <th>Kim yaradıb</th>
    <th>Tarix</th>
    <th>Həl olunub</th>
    <th>Nəticə</th>
    <th>İşləmlər</th>
  </tr>
  {% for problem in page_obj %}
  <tr>
    <td><a href="{% url 'view_problem' problem.id %}">{{ problem.name }}</a></td>
    <td><a href="{% url 'view_category' problem.category.id %}">{{ problem.category.name }}</a></td>
    <td><a href="{% url 'view_teacher' problem.created_by.id %}">{{ problem.created_by.name }}</a></td>
    <td>{{ problem.created_at }}</td>
    <td>{{ problem.solved }}</td>
    <td>{{ problem.result }}</td>
    <td>
      <a href="{% url 'edit_problem' problem.id %}">Redakte et</a> |
      <a href="{% url 'delete_problem' problem.id %}">Sil</a>
    </td>
  </tr>
  {% endfor %}
</table>
|

```

Şəkil 2.9 " Bütün problemlər " bloku

Bu sətirlərdə " Bütün problemlər " başlığı ilə bölmə göstərilir. Buraya yeni problem əlavə etmək üçün keçid də daxildir.

Problem siyahısının strukturu üçün <table> elementindən istifadə olunur. Cədvəl başlığı sırası üçün <tr> elementinə malikdir və onun daxilində <th> elementləri sütun başlıqlarını təmsil edir.

Cədvəl sütunları aşağıdakı kimi müəyyən edilir:

"Adı" : Problemin adını ifadə edir. O, parametr kimi problem identifikatoru ilə 'view_problem' adlı URL modelindən istifadə edərək problem təfərrüatları səhifəsinə işarə edən keçidi ehtiva edir.

"Kateqoriya": Problemin kateqoriyasını təmsil edir. O, parametr kimi kateqoriya ID ilə 'view_category' adlı URL modelindən istifadə edərək kateqoriya təfərrüatları səhifəsinə işarə edən keçid daxildir.

“Kim yaradıb”: Problemi yaradanı təmsil edir. Buraya parametr kimi müəllim ID-si ilə 'view_teacher' adlı URL nümunəsindən istifadə edərək müəllim təfərrüatları səhifəsinə işarə edən keçid daxildir.

"Tarix": Problemin yaranma tarixini bildirir.

"Həl olunub": Problemin həll olunub-olunmadığını bildirir.

“Nəticə”: Problemin nəticəsini ifadə edir.

"İşləmlər": Problemlə bağlı yerinə yetirilə biləcək hərəkətləri təmsil edir. Buraya problemi redaktə etmək və problemi silmək üçün keçidlər daxildir. Bu hərəkətlər üçün URL-lər parametr kimi problem ID-si ilə 'edit_problem' və 'delete_problem' adlı URL nümunələrindən istifadə etməklə yaradılır.

Cədvəl daxilində döngə təkrarlanır

PROBLEM STATICS

Təqdim olunan kod parçası 'base.html' adlı əsas şablonu genişləndirən Django şablonudur. O, müəllimlər cədvəlini göstərmək üçün HTML işarələməsini ehtiva edən "məzmun" adlı bloku müəyyən edir.

Kodu parçalayaq:

```
{% extends 'base.html' %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq 'base.html' şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir “content” adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```

<h1>Müəlimlərin siyahısı</h1>
<a href="{% url 'add_teacher' %}">Müəllim əlavə et</a>
<table>
  <thead>
    <tr>
      <th>Adı</th>
      <th>Fənn</th>
      <th>Problemlər</th>
      <th>İşləmlər</th>
    </tr>
  </thead>
  <tbody>
    {% for teacher in teachers %}
    <tr>
      <td><a href="{% url 'view_teacher' teacher.id %}">{{ teacher.name }}</a></td>
      <td>{{ teacher.subject }}</td>
      <td>{{ teacher.problem_set.count }}</td>
      <td><a href="{% url 'view_teacher' teacher.id %}">Baxmaq</a></td>
    </tr>
    {% endfor %}
  </tbody>
</table>

```

Şəkil 2.10 " Müəllimlər siyahısı " bloku

Bu sətirlərdə "Müəlimlərin siyahısı" başlığı ilə bölmə göstərilir. O, həmçinin yeni müəllim əlavə etmək üçün keçidi ehtiva edir.

Müəllim siyahısının strukturu üçün <table> elementindən istifadə olunur. Onun cədvəl başlığı üçün sətir (<tr>) ehtiva edən <thead> elementi var və onun daxilində <th> elementləri sütun başlıqlarını təmsil edir.

Cədvəl sütunları aşağıdakı kimi müəyyən edilir:

"Adı": Müəllimin adını ifadə edir. Buraya parametr kimi müəllim ID-si ilə 'view_teacher' adlı URL nümunəsindən istifadə edərək müəllim təfərrüatları səhifəsinə işarə edən keçid daxildir.

"Fənn": Müəllimin mövzusunun və ya intizamını təmsil edir.

"Problemlər": Müəllimlə bağlı problemlərin sayını əks etdirir. Problemlərin sayını tapmaq üçün problem_set.count istifadə edir.

"İşləmlər": Müəllimin üzərində edilə bilən hərəkətləri ifadə edir. Buraya müəllim təfərrüatlarına baxmaq üçün keçid daxildir. Bu fəaliyyət üçün URL parametr kimi müəllim ID ilə "view_teacher" adlı URL modelindən istifadə etməklə yaradılır.

Cədvəl daxilində döngə (müəllimlərdə müəllim üçün {% %}) müəllimlərin (müəllimlərin) siyahısını təkrarlayır və müvafiq sütunları müəllimin məlumatları ilə dolduraraq hər bir müəllim üçün cədvəl cərgəsi yaradır.

Nəhayət, {% endblock %} teqi "content" blokunun sonunu göstərir.

```
{% extends 'base.html' %}

{% block content %}
<h1>Статистика по проблемам</h1>
<p>Количество проблем: {{ problem_count }}</p>

<h2>Учителя</h2>
<ul>
    {% for teacher in teachers %}
    <li>{{ teacher.name }} ({{ teacher.problem_count }})</li>
    {% endfor %}
</ul>
{% endblock %}
```

Bu kod əsas şablonu {% extensions 'base.html' %} ilə genişləndirir. Məzmun blokunun daxilində ({% block content %}), "Problem statistikasını" başlığı və problemlərin sayı ({{ problem_count }}) göstərilir.

Sonra "Müəllimlər" başlığı (<h2>Müəllimlər</h2>) və müəllimlərin siyahısı (...) göstərilir. For döngəsi müəllimlər dəyişənindəki müəllimlər üzərində təkrarlanır və hər bir müəllim üçün müəllimin adı ({{ müəllim.adı }}) və problemlərin sayı ({{ müəllim.problem_count }}) <li-yə bükülür. >....

Məzmun bloku daha sonra {% endblock %} ilə bağlanır.

BASE.

Təqdim etdiyiniz kod Django veb proqramında baxışlar toplusudur. Bu görünüşlər müxtəlif HTTP sorğularını idarə edir və müxtəlif əməliyyatları yerinə yetirmək üçün modellər və formalarla qarşılıqlı əlaqə yaradır.

Baxışların və onların funksiyalarının bölgüsü:

indeks (sorgu): Problem sayı, kateqoriya sayı, müəllim sayı və ən son 10 problemin siyahısı kimi ümumi statistika ilə indeks səhifəsini təqdim edir.

add_category(request): Yeni kateqoriyanın yaradılmasını idarə edir. Kateqoriya əlavə etmək üçün forma təqdim edir və uğurlu təqdim edildikdən sonra kateqoriya siyahısı səhifəsinə yönləndirir.

list_categories(sorgu): Bütün kateqoriyaları götürür və onlara ümumi problem sayı və həll edilmiş problem sayı ilə annotasiya edir. Kateqoriya siyahısı səhifəsini göstərir.

view_category(sorgu, kateqoriya_id): İdentifikatoru ilə xüsusi kateqoriyanı alır və əlaqəli problemlər də daxil olmaqla onun təfərrüatlarını göstərir.

add_teacher(request): Yeni müəllimin yaradılmasını idarə edir. Müəllim əlavə etmək üçün forma təqdim edir və müvəffəqiyyətlə təqdim edildikdən sonra müəllim siyahısı səhifəsinə yönləndirir.

list_teachers(request): Bütün müəllimləri götürür və müəllim siyahısı səhifəsini təqdim edir.

view_teacher(request, teacher_id): Müəyyən bir müəllimi şəxsiyyət vəsiqəsi ilə götürür və əlaqəli problemlər də daxil olmaqla, onların təfərrüatlarını göstərir.

add_problem(sorgu): Yeni problemin yaradılmasını idarə edir. Problem əlavə etmək üçün formanı təqdim edir və uğurlu təqdim edildikdən sonra problem siyahısı səhifəsinə yönləndirir.

list_problems(request): Bütün problemləri əldə edir, onları səhifələşdirir (hər səhifədə 10 problem) və problem siyahısı səhifəsini səhifələşdirmə ilə göstərir.

view_problem(sorgu, problem_id): Müəyyən problemi ID-si ilə götürür və onun təfərrüatlarını göstərir.

filter_problems(sorgu): İstifadəçi tərəfindən müəyyən edilmiş meyarlar əsasında problemlərin süzülməsini idarə edir. Filtr parametrlərini daxil etmək üçün formanı təqdim edir, süzülmüş problemləri əldə edir, onları səhifələşdirir və problem siyahısı səhifəsini süzgəcdən keçmiş nəticələrlə göstərir.

sil_problem(istək, problem_id): Müəyyən bir problemi identifikatoru ilə silir və problem siyahısı səhifəsinə yönləndirir.

problem_statistikası(sorğu): Problem statistikasını, o cümlədən problemin ümumi sayı və onların problemlərinin sayına görə sıralanmış müəllimlərin siyahısını alır. Problem statistikası səhifəsini təqdim edir.

edit_problem(sorğu, problem_id): Problemin redaktəsini idarə edir. İdentifikatoru ilə müəyyən bir problemi götürür, problemin məlumatları ilə əvvəlcədən doldurulmuş formanı təqdim edir və forma təqdim edildikdən sonra problemi yeniləyir. Uğurlu yeniləmədən sonra problem siyahısı səhifəsinə yönləndirilir.

Bu görünüşlər Kateqoriya, Problem, ProblemFilter və Müəllim kimi modellərlə, eləcə də bu modellərin nümunələrini yaratmaq və redaktə etmək üçün müvafiq formalarla qarşılıqlı əlaqədə olur. Onlar həmçinin müxtəlif Django funksiyalarından istifadə edirlər ki, bunlar saymaq, süzgəcdən keçirmək, səhifələndirmək və identifikatorları ilə obyektləri əldə etməkdir.

III FƏSİL. MƏKTƏBLƏRİN YARANAN PROBLEMLƏRİN PYTHONDA ANALİZİ

3.1. İdarəetmədə yaranan problemlər hansılardır ?

Məktəblərin və təhsil müəssisələrinin idarəetmə prosesində bir sıra problemlər ortaya çıxar. İdarəçilər bu problemlərlə başa çıxmağa çalışmalı və effektiv idarəçilik tədbirləri

almaq üçün uyğun həll yolları tapmalıdırlar. İdarəçilikdə sıx rastlanılan bir neçə problem nümunələri aşağıda verilmişdir:

İş birliyi və kommunikasiya problemi: İdarəçilər, müəllimlər, əməkdaşlar və tələbələr arasında kommunikasiya problemi yaşanması idarəçiliyin effektivliyini mənfi şəkildə təsir edə bilər. Etkili kommunikasiya və iş birliyi yoxlaması, informasiyanın paylaşılması və fikirlərin qaynaqlaşdırılması önəmlidir.

Resurs idarəsi: İdarəçilərə məhdudiyətlərlə başa çıxmaq, effektiv şəkildə resursları idarə etmək və müvafiq bütçələmə və təmin etmə proseslərini yerinə yetirmək üçün səmərəli resurs idarəsi tələb olunur.

Performans və qiymətləndirmə: İdarəetmə prosesində performansın qiymətləndirilməsi, hədəflərə nail olunmasının izlənməsi və performansın yaxşılaşdırılması problemləri ortaya çıxı bilər. Effektiv performans qiymətləndirilməsi sisteminin tətbiq edilməsi və müəllimlərin və əməkdaşların inkişafı üçün dəstək tədbirlərinin alınması önəmlidir.

Məşğəlliklərin idarə edilməsi: İdarəçilər tələbələrin məşğəlliklərini idarə etmək, uyğun fəaliyyətləri təşkil etmək və tələbələr üçün əyləncəli və məhsuldar məşğəlliklər təmin etmək məsələləri ilə üzləşə bilərlər.

İnzibati məsələlər: İdarəçilər tərəfindən həll edilməsi və qaydaların və tələbatların uyğun şəkildə yerinə yetirilməsi tələb olunan inzibati məsələlər ortaya çıxı bilər. Bunlar arasında sənədləşdirmə, məlumat idarəetməsi, icra və administrasiya proseslərinin təşkili daxildir.

İnkişaf və təlim: İdarəçilər məktəbdə inkişaf tədbirləri təşkil etmək, müəllimlərə və əməkdaşlara təlim tədbirləri almaq, müəllimlərin və əməkdaşların karyera inkişafına dəstək vermək üçün qaydalar tətbiq etməlidirlər.

Qaydaların və standartların yerinə yetirilməsi: Məktəblərdə qanunvericilik və təlim standartlarının düzgün şəkildə yerinə yetirilməsi problemi ortaya çıxı bilər. İdarəçilər bu qaydaları izləmək üçün tədbirlər almalı və tələbələrin və əməkdaşların hüquqlarının qorunmasını təmin etməlidirlər.

Qeyri-adekvat Maliyyələşdirmə: Məktəb idarəçiliyində ən çox rast gəlinən problemlərdən biri qeyri-adekvat maliyyələşmədir. Bu, çox vaxt maliyyə, texnologiya və işçi heyəti də daxil olmaqla məhdud resurslara, o cümlədən köhnəlmiş dərsliklərə, sinif materiallarının çatışmazlığına və kadrların inkişafı üçün kifayət qədər maliyyələşməyə səbəb olur. Bu,

yüksək keyfiyyətli təhsil vermək və tələbə ehtiyaclarını idarə etməkdə çətinliklərə səbəb ola bilər.

Kadr çatışmazlığı: Məktəblərin üzləşdiyi digər problem ixtisaslı müəllim çatışmazlığıdır. Bu, siniflərin sayının artmasına, ixtisaslaşdırılmış təlimatın olmamasına və mövzu üzrə təcrübə çatışmazlığına səbəb ola bilər.

Valideynlərin iştirakının olmaması: Məktəblər çox vaxt valideynləri təhsil prosesinə cəlb etmək üçün mübarizə aparır ki, bu da davamiyyətin aşağı olmasına, aşağı akademik performans və davranış problemlərinin artmasına səbəb ola bilər.

Şagird Davranışı: Şagird davranışı zorakılıq, dərstdən yayınma, narkotik istifadəsi və zorakılıq kimi məsələlər də daxil olmaqla, məktəb idarəçiliyində əhəmiyyətli problem ola bilər.

Texnologiyaya inteqrasiya: Məktəblər tələbələrə 21-ci əsrin əsas bacarıqlarını təmin etmək üçün texnologiyaları sinifə inteqrasiya etməlidir. Bununla belə, texnologiyanın inteqrasiyası idarəetmə problemi ola bilər, o cümlədən müəllimlərin texnologiyadan səmərəli istifadə etmək üçün təlim keçməsinə və tələbələrin texnologiya resurslarına bərabər çıxışını təmin etmək.

Kurikulumun inkişafı: Məktəblər dəyişən təhsil standartlarına və tələbə ehtiyaclarına cavab vermək üçün öz kurikulumlarını hazırlamalı və ona yenidən baxmalıdırlar. Bununla belə, kurikulumun hazırlanması çətin və vaxt aparan, əhəmiyyətli resurslar və təcrübə tələb edə bilər.

Müəllimlərin Hazırlığı və İnkişafı: Təhsilin keyfiyyəti müəllimlərin bacarıq və biliklərindən çox asılıdır. Məktəblər müəllimlərin öz bacarıq və biliklərini təkmilləşdirmək üçün davamlı peşəkar inkişaf imkanlarını təmin etməlidir.

Tələbə Müxtəlifliyi: Məktəblər daha müxtəlifləşdikcə, hər bir şagirdin unikal ehtiyaclarını qarşılamaq çətin ola bilər. Buraya əlilliyi olan tələbələrin, ingilis dilini öyrənənlərin və müxtəlif mədəniyyətlərdən olanların yerləşdirilməsi daxildir.

Yekun olaraq qeyd edək ki, bunlar məktəblərin öz fəaliyyətlərini idarə edərkən üzləşdiyi bəzi çətinliklərdir. Məktəblər bu problemləri müəyyən etməli və onları effektiv həll etmək üçün strategiyalar hazırlamalıdırlar. Bu, həll yollarını müəyyən etmək və arzu olunan

nəticələrə nail olmaq üçün resursları səmərəli şəkildə bölüşdürmək üçün maraqlı tərəflərlə birgə işləməyi əhatə edir.

3.2. Problemlərin Pythonda analizi

PROQRAM TƏMİNATI

Proqram təminatı kompüter sistemində xüsusi tapşırıqları yerinə yetirmək üçün nəzərdə tutulmuş təlimatlar və ya proqramlar toplusuna aiddir. Buraya əməliyyat sistemləri, proqramlar, yardım proqramları və kompüterin işləməsini təmin edən digər proqramlar daxil ola bilər. Əməliyyat sistemləri kompüter sisteminin resurslarını və fəaliyyətlərini idarə edən

və digər proqramların işləməsi üçün platforma təmin edən proqram təminatıdır. Əməliyyat sistemlərinə misal olaraq Windows, macOS və Linux daxildir.

Proqramlaşdırma dilləri həm də proqram təminatı hesab olunur, çünki onlar tərtibatçılara yeni proqram təminatı tətbiqləri və proqramları yaratmaq üçün bir vasitə təqdim edir. Proqramlaşdırma dillərinə misal olaraq Java, Python, C++ və JavaScript-i göstərmək olar.

Qeyd etdiyimiz problemləri Python vasitəsilə analiz etmək edəcəyik. Pythonla tanış olub onun işləmə qaydalarına və işlətdiyimiz Django proqramlaşdırma dilinə baxaq.

Django, Python proqramlaşdırma dili ilə yazılan və açıq qaynaq kodlu bir web çatdırıcısıdır. Django, kompleks web tətbiqləri və saytların sürətlə və asanlıqla inkişaf edilməsini təmin edən bir framework-dür. Django, MVC (Model-View-Controller) məntiqi əsasında inşa edilmişdir və bir çox geliştiricilərə effektiv və səmərəli bir şəkildə web tətbiqləri yaratmağa kömək edir. Django tərəfindən təmin edilən güclü funksiyonlar və alətlər sayəsində, proqramçılar effektiv kod yazmaq, verilənlər bazaları ilə əlaqə qurmaq, istifadəçi autentifikasiyası, qruplaşdırma və daha bir çox funksiyaları rahatlıqla əlavə etmək imkanına malik olurlar.

Django'nun əsas xüsusiyyətləri aşağıdakılardır:

- MVC arxitekturası: Django, Model, View və Controller (MVC) arxitekturasına əsaslanır və tətbiqin elementlərini asanlıqla bir-biri ilə əlaqələndirir.
- Verilənlər bazaları ilə əlaqə: Django, çeşitli verilənlər bazaları ilə əlaqə qurmağı dəstəkləyir və ORM (Object-Relational Mapping) vasitəsi ilə verilənlər bazasından məlumatları manipulyasiya etmək üçün asan interfeys təmin edir.
- Formlar və oturacaq idarəsi: Django, forma yaratmaq və oturacaq idarəsi üçün səmərəli funksionallığı daxil edir. İstifadəçilər tərəfindən daxil edilən məlumatları yoxlamaq, işləmək və validasiya etmək üçün asanlıqla istifadə edilən alətlər təklif edir.
- İstifadəçi autentifikasiyası və yetkiləndirmə: Django, istifadəçi autentifikasiyası və yetkiləndirməni dəstəkləyir. İstifadəçilərin giriş etməsi, şifrənin təsdiqlənməsi və yetkiləndirilmiş istifadəçilərin məhdudiyətlərini təyin etmək üçün inkişaf etmiş funksionallığı mövcuddur.

- Admin paneli: Django ilə tətbiqinizə inkişaf etdikdən sonra admin paneli təmin edə bilərsiniz. Bu panel, tətbiqinizin idarə edilməsini, verilənlər bazasını dəyişdirməyi və qruplaşdırma funksiyalarını asanlaşdırır.
- Django, güclü və zəngin funksiyonallığı, səmərəli syntax-ı və geniş təminatı ilə çoxsaylı web proqramçılarının seçimi olmuşdur. Django ilə tətbiq inkişaf etdirmək, daha sürətli və daha effektiv olmağa kömək edir.

Python, prosedur, obyekt yönümlü və funksional proqramlaşdırma da daxil olmaqla bir çox proqramlaşdırma paradigmasını dəstəkləyir. Veb inkişafı, elmi hesablama, məlumatların təhlili, süni intellekt, maşın öyrənməsi və s. kimi tapşırıqlar üçün geniş kitabxanalara və çərçivələrə malikdir.

Python platformadan müstəqildir, yəni siz bir platformada (məsələn, Windows) kod yazırsınız və onu başqa bir platformada (Linux və ya macOS kimi) heç bir dəyişiklik etmədən işlədə bilərsiniz. O, həmçinin resursları, kitabxanaları və paketləri təmin edən, öyrənməyi və istifadəni asanlaşdıran fəal və dəstəkləyici icmaya malikdir.

Ümumilikdə, Python-un sadəliyi, universallığı və icma dəstəyi onu veb inkişafı, məlumatların təhlili, elmi hesablama, avtomatlaşdırma və süni intellekt daxil olmaqla müxtəlif sahələrdə ən çox istifadə olunan proqramlaşdırma dillərindən birinə çevirmişdir.

Python ilə işləmək üçün bəzi əsas qaydalar bunlardır :

1. Python kod bloklarını müəyyən etmək üçün girintidən istifadə edir, ona görə də ardıcıl və düzgün girinti saxlamaq çox vacibdir. Tipik olaraq, hər girinti səviyyəsi üçün dörd boşluq istifadə olunur.
2. PEP 8 Stil Bələdçisi: PEP 8 Python kodu üçün rəsmi üslub bələdçisidir. Bu təlimata riayət etmək kodun oxunaqlılığını və ardıcılığını qorumağa kömək edir. O, dəyişənlərin, funksiyaların və siniflərin adlandırılması üçün konvensiyaları, həmçinin kodun formatlaşdırılması üçün təlimatları əhatə edir.
3. Mənalı adlardan istifadə edilməlidir: Dəyişənlər, funksiyalar və siniflər üçün təsviri və mənalı adlar seçin. Bu, kodunuzun başa düşülməsini və saxlanmasını asanlaşdırır.

4. Şərh vermək: Müəyyən bölmələrin və ya mürəkkəb alqoritmlərin məqsədini və funksionallığını izah etmək üçün koda şərhlər daxil edilməlidir. Aydın və qısa şərhlər digər tərtibatçılara (və hətta özünüzə) kodun məqsədini başa düşməyə kömək edir.
5. Kodu modullaşdırmaq: Kodu daha kiçik, təkrar istifadə edilə bilən funksiyalara və siniflərə bölünməsi nəzərdə tutulur. Bu, kodun təkrar istifadəsini, oxunuşunu və daha asan sazlamayı təşviq edir.
6. Səhvlərin idarə edilməsi: try-except bloklarından istifadə edərək koda müvafiq səhvlərin idarə edilməsi mexanizmlərini tətbiq etmək. Bu, gözlənilməz qəzaların qarşısını almağa kömək edir və səhvləri zərif şəkildə idarə etməyə imkan verir.
7. Oxunma qabiliyyəti: Python kodu asan oxunmalı və başa düşülməlidir. Mənalı dəyişən və funksiya adlarından istifadə edib, yığcam və ifadəli kod yazmaq və “Oxunma qabiliyyəti sayır” prinsipinə əməl edilməsi.
8. Sənədləşdirmə: Funksiyaların, siniflərin və modulların təsvirlərini təmin etmək üçün docstrings istifadə edərək kodunuzu sənədləşdirin. Bu sənədlər avtomatik olaraq yaradıla bilər və kodunuzun istifadəçiləri və baxıcıları üçün faydalı istinad kimi xidmət edir.
9. Test: Düzgünlüyünü və davamlılığını təmin etmək üçün kodunuz üçün testlər yazın. Python icmasında test paketləri yaratmağa kömək edə bilən unittest, pytest və doctest kimi bir neçə sınaq çərçivəsi var.
10. Python İdiomlarından istifadə edilməsi: Python-un öz idiomatik üslubu və işlərin görülmə üsulu var. Pythonic idiomlarını qəbul edib və daha səmərəli və qısa kod yazmaq üçün mümkün olduqda daxili funksiyalardan və kitabxanalardan istifadə etmək lazımdır.

Bu qaydalar Python kodumuzun oxunaqlı olmasını, saxlanmasını və Python icması daxilində ən yaxşı təcrübələrə əməl etməsini təmin etməyə kömək edir. Bu qaydalara riayət etmək kodumuzla işləməyimizi asanlaşdıracaq, səhvləri azaldacaq.

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Category, Problem, ProblemFilter, Teacher
from .forms import CategoryForm, TeacherForm, ProblemFilterForm, ProblemForm
from django.db.models import Count
from django.core.paginator import Paginator
from django.db.models import Count, Q

def index(request):
    problem_count = Problem.objects.count()
    category_count = Category.objects.count()
    teacher_count = Teacher.objects.count()
    problems = Problem.objects.all()[:10] # limit to 10 problems
    categories = Category.objects.annotate(
        problem_count=Count("problem")
    ) # get problem count for each category
    context = {
        "problem_count": problem_count,
        "category_count": category_count,
        "teacher_count": teacher_count,
        "problems": problems,
        "categories": categories,
    }
    return render(request, "myapp/index.html", context)

def add_category(request):
    if request.method == "POST":
        form = CategoryForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect("list_categories")
```

```
else:
    form = CategoryForm()
    return render(request, "myapp/add_category.html", {"form": form})

def list_categories(request):
    categories = Category.objects.annotate(
        problem_count=Count("problem"),
        solved_problem_count=Count("problem", filter=Q(problem__solved=True)),
    )
    return render(request, "myapp/list_categories.html", {"categories": categories})

def view_category(request, category_id):
    category = get_object_or_404(Category, id=category_id) # get category or 404
    problems = category.problem_set.all() # get problems for this category
    return render(
        request,
        "myapp/view_category.html",
        {"category": category, "problems": problems},
    )

def add_teacher(request):
    if request.method == "POST":
        form = TeacherForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect("list_teachers")
    else:
        form = TeacherForm()
    return render(request, "myapp/add_teacher.html", {"form": form})
```



```
def list_teachers(request):
    teachers = Teacher.objects.all()
    return render(request, "myapp/list_teachers.html", {"teachers": teachers})

def view_teacher(request, teacher_id):
    teacher = get_object_or_404(Teacher, id=teacher_id) # get teacher or 404
    problems = teacher.problem_set.all() # get problems for this teacher
    return render(
        request, "myapp/view_teacher.html", {"teacher": teacher, "problems": problems}
    )

def add_problem(request):
    if request.method == "POST":
        form = ProblemForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect("list_problems")
    else:
        form = ProblemForm()
    return render(request, "myapp/add_problem.html", {"form": form})

def list_problems(request):
    problems = Problem.objects.all()
    paginator = Paginator(problems, 10) # paginate by 10 problems per page
    page_number = request.GET.get("page")
    page_obj = paginator.get_page(page_number)
    return render(request, "myapp/list_problems.html", {"page_obj": page_obj})
```

```
def view_problem(request, problem_id):
    problem = get_object_or_404(Problem, id=problem_id) # get problem or 404
    return render(request, "myapp/view_problem.html", {"problem": problem})

def filter_problems(request):
    if request.method == "POST":
        form = ProblemFilterForm(request.POST)
        if form.is_valid():
            problem_filter = form.save(commit=False) # create instance of ProblemFilter
            problems = problem_filter.get_problems() # get filtered problems
            paginator = Paginator(problems, 10) # paginate by 10 problems per page
            page_number = request.GET.get("page")
            page_obj = paginator.get_page(page_number)
            return render(
                request,
                "myapp/list_problems.html",
                {"page_obj": page_obj},
            )
        else:
            form = ProblemFilterForm()
    return render(request, "myapp/filter_problems.html", {"form": form})

def delete_problem(request, problem_id):
    problem = get_object_or_404(Problem, id=problem_id) # get problem or 404
    problem.delete()
    return redirect("list_problems")

def problem_statistics(request):
    problem_count = Problem.objects.count()
    teacher_count = (
        Teacher.objects.filter(problem__isnull=False)
```

```

        Teacher.objects.filter(problem__isnull=False)
        .annotate(problem_count=Count("problem"))
        .order_by("-problem_count")
    ) # get teachers with at least one problem and order by problem count
    return render(
        request,
        "myapp/problem_statistics.html",
        {"problem_count": problem_count, "teacher_count": teacher_count},
    )

def edit_problem(request, problem_id):
    problem = get_object_or_404(Problem, id=problem_id)
    if request.method == "POST":
        form = ProblemForm(request.POST, instance=problem)
        if form.is_valid():
            form.save()
            return redirect("list_problems")
    else:
        form = ProblemForm(instance=problem)
    return render(request, "myapp/edit_problem.html", {"form": form})

```

Django şablon dilində yazılmış kod parçasını təqdim etmişik. Bu kod 'base.html' adlı əsas şablonu genişləndirir və onun daxilində 'content' adlı bloku müəyyən edir.

"Məzmun" blokunda kateqoriya əlavə etmək üçün HTML forması var. Kodu parçalayaq: Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

Bu sətirdə "Kateqoriyaya əlavə et" mətnini əks etdirən HTML başlıq elementi (<h1>) var .

```
<h1>Kateqoriya əlavə etmək</h1>
```

Bu kod parçası HTML formasını təmsil edir. Metod atributu forma məlumatlarının HTTP POST sorğusu vasitəsilə göndəriləcəyini göstərən "yazıya" təyin edilib. {% csrf_token %} şablon teqi təhlükəsizlik məqsədləri üçün formada CSRF (Saytlararası Sorğu Saxtakarlığı) tokeninin daxil edilməsini təmin edir.

Formaya form.name adı ilə sahə üçün etiket daxildir. {{ form.name }} dəyişəni kateqoriya adını daxil etmək üçün giriş sahəsini təqdim edir. type="submit" atributu "Əlavə et" etiketi ilə göndərmə düyməsi yaradır.

```
<form method="post">
  {% csrf_token %}
  <label for="{{ form.name.auto_id }}">Kateqirynın adı :</label>
  {{ form.name }}
  <input type="submit" value="Добавить">
</form>
```

Bu sətir “məzmun” blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və kateqoriya əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir və istifadəçilərə kateqoriya adını daxil edib onu təqdim etməyə imkan verir.

```
{% endblock %}
```

Bu sətir “məzmun” blokunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və kateqoriya əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir və istifadəçilərə kateqoriya adını daxil edib onu təqdim etməyə imkan verir.

```
1  {% extends "base.html" %}
2
3  {% block content %}
4  <div class="container">
5    <h1>Problem əlavə etmək</h1>
6    <form method="POST" action="{% url 'add_problem' %}">
7      {% csrf_token %}
8      {{ form.as_p }}
9      <button type="submit" class="btn btn-primary">Təsdiqlə</button>
10   </form>
11   {% if form.errors %}
12   <div class="alert alert-danger" role="alert">
13     {{ form.errors }}
14   </div>
15   {% endif %}
16 </div>
17 {% endblock %}
```

Təqdim olunan kod parçası Django şablon dilində yazılmışdır. O, "base.html" adlı əsas şablonu genişləndirir və "content" adlı bloku müəyyənləşdirir.

Kodu parçalayaq:

```
{% extends "base.html" %}
```

Bu sətir cari şablonun onun strukturunu və məzmununu miras alaraq "base.html" şablonunu genişləndirdiyini göstərir.

```
{% block content %}
```

Bu sətir "content" adlı bloka başlayır. Bloklar bu şablonu genişləndirən uşaq şablonlarında ləğv edilə bilən bölmələri müəyyən etmək üçün istifadə olunur.

```
<div class="container">
  <h1>Problem əlavə etmək</h1>
  <form method="POST" action="{% url 'add_problem' %}">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" class="btn btn-primary">Təsdiqlə</button>
  </form>
  {% if form.errors %}
  <div class="alert alert-danger" role="alert">
    {{ form.errors }}
  </div>
  {% endif %}
</div>
z
```

"Məzmun" blokunda problem əlavə etmək üçün HTML forması var. Budur onun elementlərinin bölgüsü:

`<div class="container">`: Bu div elementi üslub məqsədləri üçün "konteyner" sinfinə malikdir.

`<h1>Problemin əlavə edilməsi</h1>`: Bu sətirdə "Problem əlavə etmək" başlığı göstərilir.

`<form metodu="POST" action="{% url 'add_problem' %}">`: Bu sətir HTTP metodu "POST" olaraq təyin edilmiş HTML formasını müəyyən edir. Formanın fəaliyyət atributu forma məlumatlarının təqdim ediləcəyi URL-i müəyyən edir. `{% url 'add_problem' %}` şablon teqi "add_problem" adlı URL nümunəsi əsasında dinamik olaraq URL yaratmaq üçün istifadə olunur.

`{% csrf_token %}`: Bu şablon teqi təhlükəsizlik məqsədləri üçün formada CSRF (Saytlararası Sorğu Saxtakarlığı) tokeninin daxil edilməsini təmin edir.

`{{ form.as_p }}`: Bu dəyişən forma sahələrini `<label>` teqlərinə bükülmüş HTML paraqrafları (`<p>`) kimi təqdim edərək formanın sadə təsvirini yaradır.

`<button type="submit" class="btn btn-primary">Təsdiqlə</button>`: Bu sətir "Təsdiqlə" etiketli və "btn" CSS sinifləri olan təqdim düyməsini təmsil edir. üslub məqsədləri üçün "btn-primary".

`{% if form.errors %}`: Bu sətir hər hansı forma xətalının olub olmadığını yoxlayır.

`<div class="alert alert-danger" role="alert">`: Forma xətalı varsa, "alert" və "alert-danger" sinifləri olan bu div elementi göstərilir.

`{{ form.errors }}`: Bu dəyişən div elementində forma xətalını göstərir.

```
{% endblock %}
```

Bu xətt "məzmun" bloğunun sonunu göstərir.

Ümumiyyətlə, bu kod parçası əsas şablonu genişləndirən və problem əlavə etmək üçün formanı ehtiva edən Django şablonunu təmsil edir. Forma abzaslara bükülmüş sahələri ilə göstərilir və təqdim düyməsi verilir. Forma xətalı varsa, onlar xəbərdarlıq qutusunda göstərilir.

Bu dediklərimizə İU hissəsində də baxaq.

Proqrama daxil olduqda ilk öncə ana səhifəsi menyusu açılır. Aşağıda təsvirinə baxaq (Şəkil 3.1) :



Şəkil 3.1 " Əsas səhifə "

Burada “Əsas səhifə”, “Kateqoriya”, “Müəllim”, “Problem”, “Problem filtri” başlıqları əks olunub. Ondan aşağıda isə “Problem sayı”, “Kateqoriya sayı”, “Müəllim sayı” göstərilib. Digər səhifələrə daxil olaq və strukturlarına baxaq.

Kateqoriya hissəsinə keçid etsək bu zaman aşağıdakı görünüşdə olan səhifəmiz açılacaq(Şəkil 3.2) :

Əsas səhifə	Kateqoriya	Müəllim	Problem	Problem filtri
Kateqoriyaların siyahısı				
Kateqoriya əlavə et				
Adı	Problemlərin sayı	Həll olunan problem sayı	İşlmlər	
Müəllim-valideyn konflikti	1	1	Baxmaq	
Müəllim-şagird konflikti	2	1	Baxmaq	
Müəllim-rəhbərlik konflikti	5	4	Baxmaq	
Valideyn-rəhbərlik problemi	0	0	Baxmaq	
Müəllim-heyət konflikti	0	0	Baxmaq	
© 2023 Problem veb proqramı				

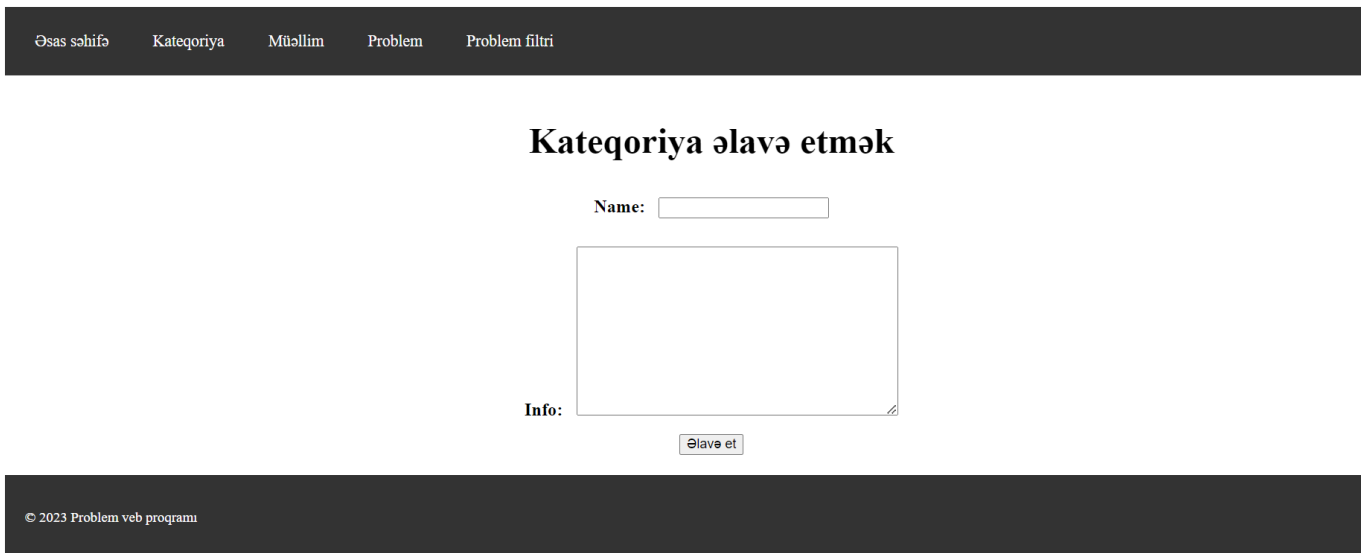
Şəkil 3.2 " Kateqoriyaların siyahısı " səhifəsi

Burada Kateqoriya siyahısı açılacaqdır. Kateqoriyanın adı, həmin kateqoriyaya daxil olan problem sayı, bu problemlərdən həll olunanların sayı və yaradılan problemlərə baxış keçirmək üçün baxış düyməsi mövcuddur.

Aşağıdakı düymə vasitəsilə yeni kateqoriyalar əlavə edə bilərik.

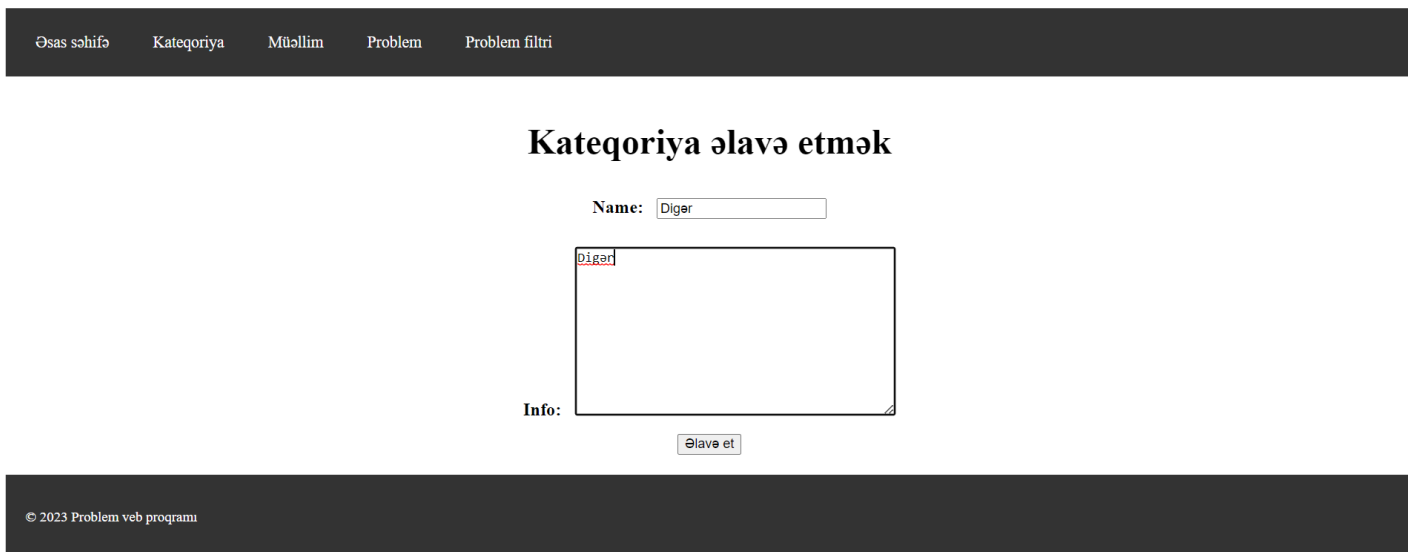
Kateqoriya əlavə et

Bu düyməyə klikləsək :



Şəkil 3.3 " Kateqoriya əlavə et" səhifəsi

(Şəkil 3.3) səhifəsi qarşımıza açılır.Burada name və info hissəsini müəyyən başlıqları qeyd etsək və əlavə et düyməsini sıxsaq bu zaman yeni kateqoriya yaranacaqdır. Baxaq (Şəkil 3.4) :



Kateqoriyaların siyahısı

Kateqoriya əlavə et

Adı	Problemlərin sayı	Həll olunan problem sayı	İşlmlər
Müəllim-valideyn konfliktli	1	1	Baxmaq
Müəllim-şagird konfliktli	2	1	Baxmaq
Müəllim-rəhbərlik konfliktli	5	4	Baxmaq
Valideyn-rəhbərlik problemi	0	0	Baxmaq
Müəllim-həyat konfliktli	0	0	Baxmaq
Digər	0	0	Baxmaq

Şəkil 3.4 "Kateqoriya siyahısı" səhifəsi

Növbəti səhifəyə keçid etsək "Müəllimlərin siyahısı" səhifəsi açılacaq (Şəkil 3.5). Burada biz müəllimlərin ad və soyadını, tədris etdiyi fənni, müəllim tərəfindən yaşanan problemləri və daxil olan işləmələrə baxış keçirə bilərik.

Müəllimlərin siyahısı

Müəllim əlavə et

Adı	Fənn	Problemlər	İşlmlər
Maryəm	Abbasova	0	Baxmaq
Zərifə Sadiqova	Tarix	1	Baxmaq
Vüsalə Məmmədova	İngilis dili	1	Baxmaq
Şəfa Cəfərli	Azərbaycan dili	0	Baxmaq
Elnur Ələsgərli	İnformatika	0	Baxmaq
Nəriman Heydərov	Kimya	1	Baxmaq
Əli Əliyev	Hərbi hazırlıq	1	Baxmaq
Sevinc Quluzadə	Fizika	0	Baxmaq

Şəkil 3.5 "Müəllimlərim siyahısı" səhifəsi

Eyni zamanda burada da digər səhifədəki kimi yeni bir müəllim əlavə etmək üçün düyməmiz var. Bu düyməmiz “Müəllim əlavə et” adlanır. Əlavə etdiyimiz müəllimlər aşağı hissəyə düşür.

Müəllim əlavə et

Nümunə üçün yeni bir müəllim əlavə etsək bu zaman aşağıdakı səhifə açılacaqdır.

(Şəkil 3.6)

Əsas səhifə Kateqoriya Müəllim Problem Problem filtri

Müəllim əlavə et

+

Name:

Subject:

Info:

© 2023 Problem veb programı

Şəkil 3.6 " Müəllim əlavə et " səhifəsi

Verilən xanaları müvafiq doldurub “Əlavə et” düyməsini klikləsək yeni qeyd etdiyimiz müəllim cədvəlimizdə əks olunacaqdır. (Şəkil 3.7)

Müəllim əlavə et

+

Name:

Subject:

Info:

Yeni əlavə etdiyimiz müəllimə cədvəlimizdə baxaq. (Şəkil 3.7)

Madlen Mehmanova	Təsviri incəsənət	1	Baxmaq
Seyidhəşim Həşimov	Fiziki tərbiyə	1	Baxmaq
Ceyran Abdullayeva	İbtidai sinif	1	Baxmaq
Gülnar Əzimova	Biologiya	0	Baxmaq
Elvin Qədirov	Həyat bilgisi	0	Baxmaq
Tural Əlizadə	Fiziki tərbiyə	0	Baxmaq
Nigar Əzimova	Fizika	0	Baxmaq
Ayşən Əlibəyli	Kimya	0	Baxmaq
Elsevər Həməzəxanzadə	İngilis dili	0	Baxmaq
Namiq	Adıgözəlov	0	Baxmaq
Elşən Ələsgərov	Tarix	0	Baxmaq

Şəkil 3.7 " Müəllimlər siyahısı" səhifəsi

Problemə daxil olduqda belə bir səhifə açılır. (Şəkil 3.8) Burada mövcud olan bütün problemləri görürük. Müəllim adı və soyadı , problemin kateqoriyası , problemi yaradan

müəllim , problemin yaranma tarixi , həll olunub – olunmaması , əgər həll olunubsa nəticəsi nəhayət sonda qeyd olunan problemlər üzərində redaktə işləri aparmaq və silmək üçün düymə qeyd olunmuşdur.

Əsas səhifə
Kateqoriya
Müəllim
Problem filtri

Bütün problemlər

Problem əlavə et

Adı	Kateqoriya	Kim yaradıb	Tarix	Həl olunub	Nəticə	İşlmlər
Müəllimə qarşı təhqir	Müəllim-valideyn konfliktli	Zərifə Sadiqova	March 3, 2023, 11:02 a.m.	True	Valideyn müəllimdən üzr istədi. Və məsələ həll edildi.	Redakte et Sil
Müəllim-rəhbərlik konfliktli	Müəllim-rəhbərlik konfliktli	Fuad Yaşar	March 13, 2023, 11:02 a.m.	True	Fuad Yaşar dəfələrlə gecikmələrinə görə izahat yazmışdır.Təkrarlandığı halda töhmət alacaqdır.	Redakte et Sil
Dərs qaydalarını pozmaq	Müəllim-şagird konfliktli	Vüsalə Məmmədova	April 4, 2023, 12:23 p.m.	True	Direktor müəvini şagirdlə danışdı və onu tənbeh etdi.Şagird müəllimdən üzr istədi və dərsə geri döndü.	Redakte et Sil

127.0.0.1:8000/list_problems/

Şəkil 3.8 "Bütün problemlər" səhifəsi

Bundan əlavə olaraq hər səhifədəki kimi burada da əlavə et butonu vardır. Yeni bir problem əlavə etmək üçün bu butona klikləyib ad,kateqoriya,müəllim adı,tarix və həll olması barədə məlumatları qeyd edirik.

Problem əlavə et

Bu zaman belə bir səhifə açılır ekranda. (Şəkil 3.9 və 3.10)

Problem əlavə etmək

Name:

Created by:

Created at:

Solved:

Measures taken:

Category:

Şəkil 3.9 "Problem əlavə et" səhifəsi

Category:

Result:

Şəkil 3.10 " Problem əlavə et" səhifəsi

Xanalara daxil olub boşluqları doldursaq və “təsdiqlə” düyməsini basmaqla əlavə etdiyimiz problem öz əksini tapacaqdır.

Problemləri filtrləyərk axtarışa vermək üçün menyudan problem filtrinə klikləyirik və açılan səhifədə parametrlərə baxaq.

Name – hissəsində axtarış üçün ad qeyd etmək üçün istifadə edilir.

Start date – hissəsində hansı aralıqdan etibarən olan problemlərə baxmaq istəyiriksə həmin tarixi qeyd edirik.

End date – hissəsində isə hansı aralığa qədərki tarixi qeyd edirik.

Category – hansı kateqoriyadakı problemlərə baxmaq istəyiriksə seçirik.

Teacher – burada isə hansı müəllimi axtarırsaq o müəllimi seçib qeyd edirik

Və sonda əgər problem həll edilibsə solved seçirik , əgər hələ həll edilməyibsə bu zaman xananı boş qoyaraq filtrləyirik. Nəhayət ekrana axtardığımız problemlər çıxır.

Filter Problems

Name:

Start date:

End date:

Category:

Teacher:

Solved:

Nümunə üçün filtri işə salaq :

Filter Problems

Name:

Start date:

End date:

Category:

Teacher:

Solved:

Filt düyməsini klikləyib çıxan problemlərə baxaq (Şəkil 3.11) :

Əsas səhifə Kateqoriya Müəllim Problem Problem filtri

Bütün problemlər

Problem əlavə et

Adı	Kateqoriya	Kim yaradıb	Tarix	Həl olunub	Nəticə	İşləmlər
Müəllim-rəhbərlik konflikti	Müəllim-rəhbərlik konflikti	Fuad Yaşar	March 13, 2023, 11:02 a.m.	True	Fuad Yaşar dəfələrlə gecikmələrinə görə izahat yazmışdır. Təkrarlandığı halda töhmət alacaqdır.	Redakte et Sil

© 2023 Problem veb proqramı

Şəkil 3.11 "Axtarış üzrə bütün problemlər" səhifəsi

Axtardığımız müəllim hesabında tək bir problemimiz var bu da ekranda əks olunmuşdur.

NƏTİCƏ

Bəs belə bir sual edilir :

Bu dostlar məktəblərdə fəaliyyət göstərdikdən sonra məktəblərdə hansı dəyişikliklər qeydə alınıb və nə kimi müsbət göstəricilər var ?

Əlbəttə bu suala cavab tapmaq üçün bil əvvəl və indini müqayisə etmək kifayətdir ki, aradakı fərqi görərik. Şagirdlərin məktəbdə siqaret çəkmə halları 80%-cən azalmış, məktəbə kənar şəxslərin daxil olmasının qarşısı alınmış , şagirdlərin dərstdən qaçmaq kimi problemləri minimuma endirilmişdir. Şagirdlər artıq Məktəblinin dostları ilə dostluq edib həm onlara qarşı hörmət bəsləyir eyni zamanda onların daima Nazirliyin kamerası tərəfindən izlənməsi onlarda böyük məsuliyyətin yaranmasına səbəb olmuşdur. Layihə olan məktəblərdə şagirdlərarası davaların sayı demək olar yox dərəcəsinə düşmüşdür.

Bu dostlar təkcə məktəblilərlə dostluq etmir eyni zamanda məktəb idarəsinə də nəzarət edirlər. Məktəbə ilk gəldiklərində birinci işləri məktəbin zəlzələ,yanğın kimi hallar üçün təxliyyə çıxışlarını və yanğın balonlarını yoxlayırlar. Məktəb heyətinin təhlükəsizliyini öndən tuturlar. Daha sonra məktəbi ümumi nəzarət edib açıq cərəyan varsa bunu dərhal bildirirlər. Məktəb idarəetməsi dedikdə məktəbdə baş verən hər hansı qanun pozuntusu , qeyri-şəffaf olan məsələləri aşkarlayıb , bunu işıqlandırırırlar. Əgər davamiyyət pozuntusu olan şagird varsa bunun səbəbini araşdırmaq üçün ilk öncə sinif rəhbəri ilə danışib daha sonra valideynlə əlaqə saxlayırlar. Bacardıqları qədər ilə heç bir şagirdi gözdən yayındırmamağa ,hər birinin problemini aşkarlamağa və onların güvənini , sevgisini qazanmağa çalışırlar. Şagirdlərin onlara qarşı sevgi bəsləməsi , onlarla səmimi söhbətlər etmələri, hətta onlara bir çox hallarda sarılmaq istəmələri , onlarla dostluq münasibətlərinin danılmaz faktıdır.

Məktəb idarəsində bu layihə əməkdaşlarının fəaliyyəti müəllimlərin dərslə daha effektiv və diqqətli yanaşmasını , müəllim kollektivində daha obyektiv olmağı,təhsildə hər hansı rüşvət və bu kimi halların qarşısının alınmasına gətirib çıxarmışdır.Məktəb idarəsində direktorun da eyni şəffaflıqla işləməsi,bütün məsələlərdən agah olması,əgər onun məlumatı olmadan hər hansı hadisə baş verərsə yaxud müəyyən proses görülərsə bu zaman xəbərdar edilməsi və bu haqda tədbir görülməsinə nəzarət də layihə əməllərinə daxildir.

ƏDƏBİYYAT

1. Ağamalıyev R. Azərbaycan təhsili XXI əsrə doğru: idarəetmə, problemlər, islahatlar, Bakı, 1998
2. Abdullayev A., Məktəbin idarə edilməsinin demokra-tikləşdirilməsi, Bakı, 1993
3. Cabbarov R.H., Təhsil müəssisələrində təlim-tərbiyə işlərinin planlaşdırılması sistemi, Bakı, 1996 18. Cabbarov R.H., Təhsil müəssisələrinin idarəetmə məsələləri, Bakı, 2001
- 4.“Azərbaycan Respublikasında ümumi orta təhsilin dövlət standartlarının təsdiqi haqqında” Azərbaycan Respublikası Nazirlər Kabinetinin 26 aprel 1999-cu il tarixli 72 №-li qərarı, “Azərbaycan müəllimi” qəzeti, 13 – 19 may, 1999
5. The rules of management Richard Templar , 2011
6. Azərbaycan Respublikasında fasiləsiz pedaqoji təhsil və müəllim hazırlığının Konsepsiya və Strategiyası, Bakı, 2007
7. Azərbaycan Respublikasında təhsil sahəsində islahatlar üzrə dövlət komissiyası haqqında Azərbaycan Respublikası prezidentinin sərəncamı, Bakı, 30 mart, 1998
8. Bayramov H, Kəlbəliyev Ə., Təhsilin idarə edilməsinin 233 əsasları, Bakı, 1998
- 9.Essential Rules for School Management A.OŞFakunle -1984
10. Data Science do Zero Joel Grus
11. Kərim Tahiroğlu Python ilə proqramlaşdırma ,2016
12. Python Programming: An Introduction to Computer Science – John Zelle,2004
13. 10. Python Projects Laura Cassell, Alan Gauld, 2014
14. Python Egitim Kitabı Volkan Tasci 2022
15. Python Web Development with Django Niel George - 2016
16. Building Maintainable Software : Joost Visser, Sylvan Rigal, Rob van der Leek -2006
17. Pro Django Marty Alchin – 2009
18. Python Fundamental Hints and Tips to Kick Start Your Python Skills John McMurphytərəfindən - 2016
19. Cabbarov R.H., Təhsil müəssisələrinin idarə edilməsində metodik xidmətin təşkili və planlaşdırılması, Bakı, 2005
20. Davletşina N.V., Kimliki B.B., Klarç R.C., Rey D.İ., Demokratiya: Dövlət və cəmiyyət. «Səda» nəşriyyatı: Bakı – 1999.

21. Əhmədov N. və b. «İdarəetmənin pedaqogikası və psixologiyası» Bakı, 2000
22. Əhmədov H., «Azərbaycan məktəbi və pedaqoji fikir tarixi». “Təhsil” nəşriyyatı. Bakı 2001.