ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Жортулов Аблай Сәкенұлы

Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу

ТҮСІНДІРМЕ ЖАЗБА
дипломдық жобаға

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
ПИ кафедрасының меңгерушісі
физ.-мат. ғыл.канд, профессор
_____А.Н.Молдагулова
"24" 05 2022 ж.

ТҮСІНДІРМЕ ЖАЗБА
дипломдық жобаға

Тақырыбы: «Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу»

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы

Орындаған Жортулов А.С.

Рецензент
физ.-мат. ғыл.канд, аға оқытушы
_____ И.М. Уалиева
"23" 05 2022 ж.

Ғылыми жетекші
лектор, магистр
_____ И.Э. Алпысбай
"20" 05 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН
ПИ кафедрасының меңгерушісі
физ.-мат. ғыл.канд, профессор
_____А.Н.Молдагулова
"24" 05 2022 ж.

Дипломдық жобаны орындауға
ТАПСЫРМА

Білім алушыға Жортулов Аблай Сәкенұлы
Тақырыбы«Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу»
Академиялық мәселелер жөніндегі проректоры бұйрығының №489-ПВ "24" 12 2021 ж. шешімімен бекітілген.
Орындалған жобаның өткізу мерзімі "24" 05 2022 ж.
Дипломдық жобаның бастапқы мәліметтері:
Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:
а) қоймалық процесстер функцияларын әзірлеу;
б) деректер қорын құру;
в) А қосымшасы – техникалық тапсырма
г) Б қосымшасы – бағламалық мәтін
Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен): презентацияның 20 слайдпен берілген құжат түрінде ұсынылған.
Ұсынылған негізгі әдебиеттер: 15 пайдаланылған әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	**Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері**	**Ескерту**
1. Дипломдық жобаның жоспарын құру	16.01.2022	орындалды
2. Программалау технологияларын таңдау	23.01.2022	орындалды
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау	7.02.2022	орындалды
4. Жобаны құрып, негізгі функционалдығын жүзеге асыру	21.02.2022	орындалды
5. Жобаның функционалдығын кеңейту	4.03.2022	орындалды
6. Жобаның интерфесін жетілдіру	22.03.2022	орындалды
7. Жобаны тестілеуден өткізу	1.04.2022	орындалды
8. Түсініктемелік хатты жазу	28.04.2022	орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	**Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)**	**Қолтаңба қойылған мерзімі**	**Қолы**
Нормалық бақылаушы	Жекамбаева М.Н. PhD, қауымдастырылған-профессор	18.05.22	
Бағдарламалық бөлім	Марғұлан Қ. тех.ғыл.магистрі, лектор	18.05.22	

Ғылыми жетекші _____ Н.Э. Алпысбай

Тапсырманы орындауға алған білім алушы _____ А.С. Жортулов

Күні «17» 11 2021 ж.

АҢДАТПА

«Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу» веб-қосымшасын құру пайдаланушыларға қол жетімді заттардың нақты есебін жүргізуге, сондай-ақ қажетті сатып алу процесін жеңілдетуге көмектеседі.

Дипломдық жоба кіріспеден, үш негізгі бөлімнен және қорытындыдан тұрады:

Кіріспеде жұмыстың өзектілігі ашылады, жобаның мақсаттары мен міндеттері анықталады.

Бірінші бөлімде пәндік сала зерттеліп, терминдер мен қысқартулар анықталады.

Екінші бөлімде веб-қосымшаны әзірлеу кезіндегі пайдаланылған технологиялар сипатталған.

Үшінші бөлімде жобаның орындалуы, оның архитектурасы, интерфейсі көрсетілген. UML диаграммалары мен скриншоттар берілген.

«Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу» тұжырымдамасы барлағы 36 беттен, оның ішінде 14 суреттен тұрады. Жұмысты жазу кезінде 15 сілтемелер мен мақалалар пайдаланылды.

АННОТАЦИЯ

Создание веб-приложения «Проектирование и разработка информационной системы для обеспечения поддержки процессов закупок и учёта хранения товаров на складе» призвана помочь пользователям вести четкий учёт имеющихся предметов, а также облегчить процесс необходимых закупок.

Дипломный проект состоит из введения, трех основных разделов и заключения:

В введении раскрываются актуальность работы, определяются цели и задачи проекта.

В первом разделе производится изучение предметной области и определяются термины и сокращения.

Во втором разделе описаны использованные технологии при разработке веб-приложения

В третьем разделе расскрывается реализация проекта, его архитектура, интерфейс. Предоставлены UML диаграммы и скриншоты.

Концепция «Проектирование и разработка информационной системы для обеспечения поддержки процессов закупок и учёта хранения товаров на складе» состоит всего из 36 страниц, в том числе 14 изображений. При написании работы было использовано 15 ссылок и статьи

ANNOTATION

The creation of a web application “Design and development of an information system to support procurement process and accounting for the storage of goods in a warehouse” is designed to help users keep a clear record of available items, as well as facilitate the process of necessary purchases.

The graduation project consists of introduction, three main sections and a conclusion:

The introduction reveals the relevance of the work, defines the goals and objectives of the project.

In the first section, the subject area us studied and terms and abbreviations are defined.

The second section describes the technologies used in the development of a web application.

The third section reveals the implementation of the project, its architecture, and interface. UML diagrams and screenshots are provided.

The concept of “Design and development of an information system to support procurement process and accounting for the storage of goods in a warehouse” consists of 36 pages, including 14 images. When writing the work 15 links and articles were used

МАЗМҰНЫ

	Кіріспе	9
1	Зерттеу бөлімі	10
1.1	Пәндік облысты зерттеу	10
1.2	Терминдер мен қысқартулар	11
2	Технологиялар бөлімі	12
2.1	PyCharm IDE	12
2.2	Django фреймворкі	13
2.3	MySQL Workbench	13
3	Жоба құрылымы	15
3.1	Жобаның архитектурасы	15
3.2	UML диаграммалары	17
3.3	Деректер қорының сипаттамасы	18
3.4	Веб қосымшаның интерфейсі	19
	Қорытынды	25
	Пайдаланылған әдебиеттер тізімі	26
	А Қосымшасы. Техникалық тапсырма	27
	Б Қосымшасы. Бағдарлама мәтіні	29

КІРІСПЕ

Қойма есебі бизнес процестерді және қызметкерлердің жұмыс тиімділігін оңтайландыруға көмектеседі. Ол бірқатар маңызды міндеттерді орындайды, соның ішінде: баланстар мен қорларды бақылау, сатып алыду жоспарлау, түгендеу процесін жеңілдету, дұрыс бірліктерді табу, өнім анықтамалығын жүйелеу, қателер мен ұрлықтардың санын азайту және қойма аумағын оңтайландыру.

Дипломдық жұмыстың өзектілігі қойма есебін жүргізетін адамдар санымен байланысты. Мәселен қазіргі уақытта жеке кәсіппен айналысу өте оңай. Шағын бизнесіңізді ашу үшін жеке кәсіпкер ретінде тіркелу жеткілікті және ол үшін ЭЦҚ жеткілікті болады. Одан бөлек әлеуметтік желілер арқылы да сауда жүргізуге болады. Сонымен қатар бүгінгі күнде де көптеген мекемелерге де қоймалық есеп жүргізу қажет. Алайда бірқатары есепті қағаз жүзінде жүргізеді, ал бұл қазіргі кезде ескірген әдіс. Сондықтан қоймалық есебті жүргізуге көмектесетін құрал саудамен айналыспайтын мекелерге де қажет.

Жоғарыда айтылғандарды ескере отырып қойма есебін жеңілдететін веб-қосымша әзірлеуге сешім шығарылды. Веб-қосымша келесі функциларды атқару тиіс:

- Тауарларды сақтау, қабылдау, жою процесстерін қолдау;
- Тауарларды категориясы немесе аты арқылы тізімнен іздеу;
- Жүргізілген операциялардың тарихын сақтау;
- Тауарлар тізімі мен операциялардың тарихын жүктеу;

Дипломдық жұмыстың мақсаты – неғұрлым кең аудиторияға көмектесе алатын ыңғайлы және қолданысқа жеңіл қоймалық есебті жүргізуді жеңілдететін веб-қосымшаны әзірлеу.

1 Зерттеу бөлімі

1.1 Пәндік облысты зерттеу

Қойманы басқару ісінің пайда болуы, басқа ғылымдар секілді, ғасырлар қойнауында жатыр. Ол адамдар қоғамда өзін сезіне бастаған кезде пайда болды. Еңбек ете бастаған адам еңбек құралдарын ойлап тапты, сонымен қатар өндірістің (еңбек нәтижелерінің), яғни «қойма қорларының» есебін жүргізу қажет болды. Адамзат қоғамы дамуының алғашқы кезеңдерінде барлық ақпарат адамның санасында сақталды, өйткені адамның жады маңызды оқиғаларды есте сақтауға жеткілікті болды. Экономикалық өмірдің күрделенуімен заттарды есепке алу қажет болды. Археологтар ашқан алғашқы жазбалар б.з.б 30 ғасырға жатады. Бұл мамонттың сүйектері мен тістеріндегі ойықтар, жартастағы суреттер және т.б. Ең көне құжаттарда сызықтардың реттелген кезектесуі, бір типті таңбалардың комбинациясы (нүктелер, доғалар, түзу және толқынды сызықтар) кездеседі, бірақ есепке алу объектілерінің сапалық сипаттамалары (атауы, күні немесе сақтау мерзімі және т.б.) жоқ.

Уақыт өте келе есеп жүргізу ісі дамыды себебі қай мемлекет болмасын, қай мекеме болмасын барлығы қолындағы бар заттар туралы ақпаратты білуі тиіс еді. Жазбалар болса елден-елге және уақытқа байланысты өзгеріп отырды. Құжаттар папирусте, балшық сынықтарында, ағаш тақтайларда, жануарлардың терісінде, ағаш қабығының ішкі жағында жазылып, ыдыстарда, сауыттарда, жәшіктерде сақталған болатын.

Қазіргі кезде қоймадығы заттар туралы есеп көбінесе компьютерлерде және кітапшаларда сақталады. Цифрлық түрде де адамдар әртүрлі әдістерді қолданады. Кейбіреулер «excel» құжаттарында, кейбіреулері арнайы қосымшалар арқылы иеленген заттарын есептен өткізеді. Сонымен қатар осы шақта тауарларды қадағалау есептерін жүргізу кәсіппен айналысатын жеке тұлғаларға да қажет, себебі қоршаған ортадан өте көп ақпарат қабылданады және бүкіл процесстерді жадыда сақтау тиімсіз.

Заманауи талаптарға сай барлық мекемелерде ақпартты, жұмысты цифровизациялауда. Деректерді компьютер жадында сақтау қағаз жүзінде сақтау форматына қарағанда қауіпсіздеу және тарату, өзгерту, жою процесстерін жеңілдетеді. Алайда көптеген қойманы қадағалау, басқару қосымшалары кәсіптік есепке көбірек көңіл бөледі. Соның салдарынан мекемелердің цифровизациялануы тежелуде. Сондықтан осы дипломдық жоба кәсіппен айналысатын және коммерциялық емес мекеме болып табылатын тұтынушыларға қоймада сатып алу және сақтау процесстерін басқаруға арналған веб-қосымшаны жасауды көздейді.

1.2 Терминдер мен қысқартулар

Анықтамалар, терминдер және қысқартулар 1-кестеде көрсетілген.

1-кесте – терминдер, қысқартулар және олардың анықтамасы

Термин немесе қысқарту	Анықтама
IDE – integrated development environment	Бағдарламалық қамтаманы әзірлеу үшін пайдаланатын бағларламалық құралдар жиынтығы
Debugger	Басқа бағдарламаларда, операциялық жүйе ядроларында, SQL сұрауларында және кодтың басқа түрлерінен қателер табу процесстерін автоматтандыруға арналған компьютерлік бағдарлама
Regex – regular expressions	Тұрақты өрнектер – мәтінмен жұмыс істейтін компьютерлік бағдарламаларда қолданылады, метасимволдарды қолдануға негізделген мәтіндегі ішкі жолдарды іздеуге және өңдеуге арналған ресми тіл
Фреймворк	Бағдармалық жобаны әзірлеу процессін жеңілдететін бағдармалық қамтама
SQL – structured query language	Реляциялық дерекқордағы деректерді құру, өзгерту және өңдеу үшін қолданылатын декларативті бағдарламалау тілі
UML – Unified Modeling Language	Бағдарламалық қамтамасыз етуді әзірлеуде нысанды модельдеуге, бизнес процесстерді модельдеуге, жүйелік инженерияға және ұйымдық құрылымдарды көрсетуге арналған графикалық сипаттау тілі
Интернационализация	Бағдарламалық өнімнің кез-келген жерде пайдалануы үшін өнімнің бейімделуі
Рефакторинг	Бағдарламаның сыртқы әрекетіне әсер етпей және оның қалай істейтінін түсінуді жеңілдету мақсатында оның ішкі құрылымын өзгерту процесі

2 Технологиялар бөлімі

2.1 PyCharm IDE

PyCharm – Python бағдарламау тіліне арналған интеграцияланған даму ортасы. Ол кодты талдау құралдарын, графикалық түзеткішті, блокты тексеруді іске қосу құралын ұсынады және Django-да веб дамуды қолдайды. PyCharm Intellij IDEA негізінде JetBrains компаниясымен әзірленген. PyCharm Windows, macOS, Linux-пен үйлесімді. Екі нұсқасы бар: Community Edition Apache License лицензиясында (тегін); Professional Edition меншікті бағдарлама (ақылы).

PyCharm PyDev-пен бәсекелестік құру үшін Python-ға арналған интеграциланған даму орталары нарығына шығарылды (алайда, қазіргі уақытта PyCharm кодты түзету үшін PyDev қолданады). Бета нұсқасы 2010 жылдың шілдесінде шығарылды, 1.0 сол жылдың қарашасында шығарылды. 2.0 нұсқасы 2011 жылдың 13 желтоқсанында, 3.0 нұсқасы 2013 жылдың 24 қыркүйегінде шығарылды. PyCharm Community Edition, тегін және ашық бастапқы нұсқасы, 2013 жылдың 22 қазанында жарық көрді.

2016 жылдың наурызында JetBrains қол қою лицензиялау можеліне көшті, сонымен бірге нұсқалардың нөмірленуі де өзгерді. Енді нұсқа нөмері YYYY.R, мұндағы YYYY-шығарылған жылы, ал R-осы жылдың ішіндегі шығарылым нөмірі.

PyCharm-ның негізгі мүмкіндіктері:

- Функционалды синтаксистік бөлектеу, авто форматтау, авто шегіністері бар код редакторы.
- Қарапайым код навигациясы
- Авто толықтыру, авто импорт, код шаблондары секілді код жазу процессіне көмек.
- Редактордың терезесіндегі кез-келген элемент үшін құжаттаманы жылдам қарау, шолғыш арқылы сыртқы құжаттаманы қарау.
- Жобадағы жылдам жаһандық өзгерістерді жүзеге асыру үшін кең мүмкіндіктер беретін қуатты код рефакторингі.
- Django, Flask фреймворкторын қолдау.
- Google App Engine қолдау.
- Біріктірілген Unit тестілеу.
- Толық функциналды графикалық түзеткіш (Debugger).
- Javascript, Coffescript, HTML/CSS, SASS, LESS, HAML редакторы.

Дипломдық жұмыста PyCharm-ның Community Edition 2021.3.2 нұсқасы қолданылды.

2.2 Django фреймворкі

Django – Python бағдармалық тілінде жазылған тегін және ашық қолданыстағы веб-қосымшалар үшін фреймворк. Оны Django Software Foundation ұйымы қолдайды. Бірінші нұсқасы 2005 жылы 21 шілде күні жарық көрді. Django-ны қолданытын үлкен жобаларға Instagram, Disqus, Pinterest, Mozilla, The Washington Times және басқалары жатады.

Django-ны кез-келген сайтты құру үшін пайдалануға болады – мазмұнды басқару жүйелері мен викилерден бастап әлеуметтік желі мен жаналықтар сайттарына дейін. Ол кез-келген клиенттік платформамен жұмыс істей алады және контентті көп форматта жеткізе алады (HTML, RSS, JSON, XML). Сонымен қатар Django жобаны SQL инъекция, сайтаралық скриптинг, кликджекинг сияқты осалдықтардан қорғай алады.

Django-ның басқа мүмкіндіктері:

- Интернационализация
- Кэштеу жүйесі
- Кірістірілген администратор интерфейсі
- Regex негізіндегі URL диспетчері
- Қосымша запростарды өңдеушілерді құру үшін арналған фильтрлер жүйесі

Фреймворктың ядросына енгізілген мүмкіндіктерінен бөлек оған қосымша пакеттер бар. Дипломдық жұмыста Django 4.0.3 нұсқасы қолданылды.

2.3 MySQL Workbench

MySQL Workbench – дерекқорды жобалауды, модельдеуді, құруды және пайдалануды MySQL дерекқор жүйесі үшін біртұтас жіксіз ортаға біріктіретін визуалды мәліметтер базасын жобалау құралы. C/C++/Objective-C тілдерінде жазылған және алғашқы нұсқасы 2005 жылдың қыркүйегінде шығарылды. Бағдарлама кроссплатформалық, яғни қазіргі кездегі барлық делік операциляқ жүйелерде жұмыс істейді. MySQL Workbench үш басылымда ұсынылады: Community Edition (тегін), Standard Edition (ақылы бірақ қосымша функциялары бар), Enterprise Edition (үлкен бизнеске).

Workbench-тің мүмкіндіктері:

- Reverse Engineering – серверді бұрыннан бар дерекқордан кесте құрылымен қалпына келтіру.
- Дерекқордың моделін графикалық түрде көрсету
- Кестелер арасындағы орнатудың көрнекі және функционалды механизмі
- Кестедегі деректерді визуалды режимде өңдеу мүмкіндігі.
- Ыңғайлы SQL запростар редакторы, оларды бірден серверге жіберуге және жауап ретінде кесте түрінде алуға мүмкіндік береді.

Дипломдық жұмыста MySQL Workbench 8.0 қолданылды.

3 Жоба құрылымы

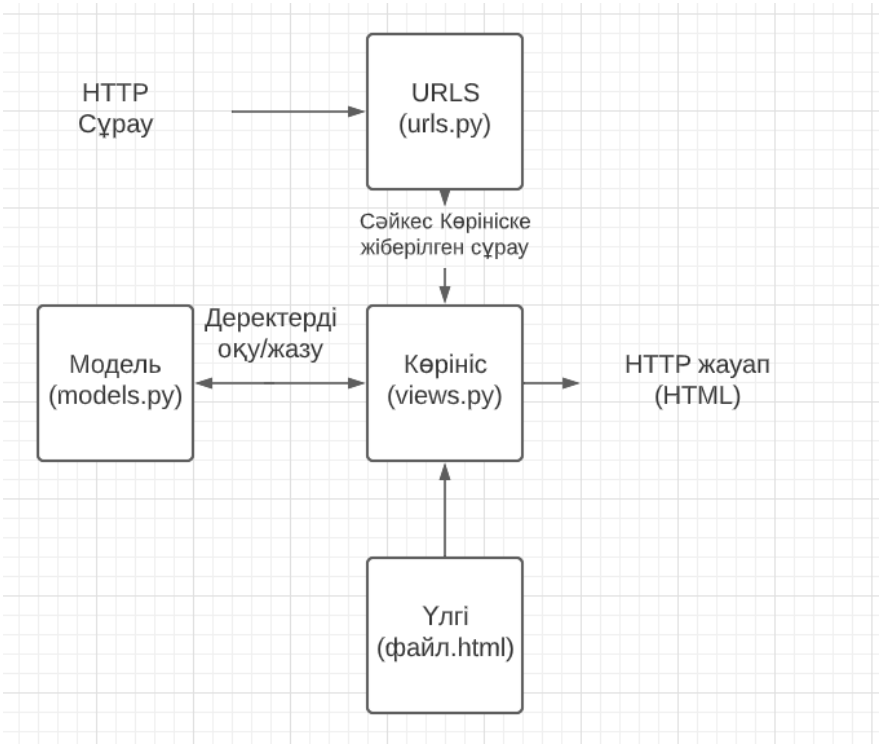
3.1 Жобаның архитектурасы

Django-ның архитектурасы Model-View-Controller (MVC) сияқты. Классикалық MVC үлгісінің контролері Django Көрініс (View) деп атайтын деңгейге сәйкес келеді және Көріністің презентация логикасы Django-да Үлгі (Template) қабаты арқылы жүзеге асады. Осыған байланысты Django-ның қабатты архитектурасы жиі «Модель-Үлгі-Көрініс» (Model-Template-View MTV) деп аталады.

Django құжаттамасында Модельді «деректердің негізгі өрістері мен әрекетін қамтитын деректер туралы ақпарат көзі» ретінде анықтайды. Әдетте бір модель дерекқордағы бір кестені көрсетеді. Модельдер деректер туралы ақпаратты сақтайды. Бұл деректер атрибуттармен немесе өрістермен ұсынылған. Модель қарапайым класс болғандықтан, ол Django-ның басқа деңгейлер туралы ештеңе білмейді. Деңгейлер арасында өзара әрекеттесу API арқылы жүзеге асады.

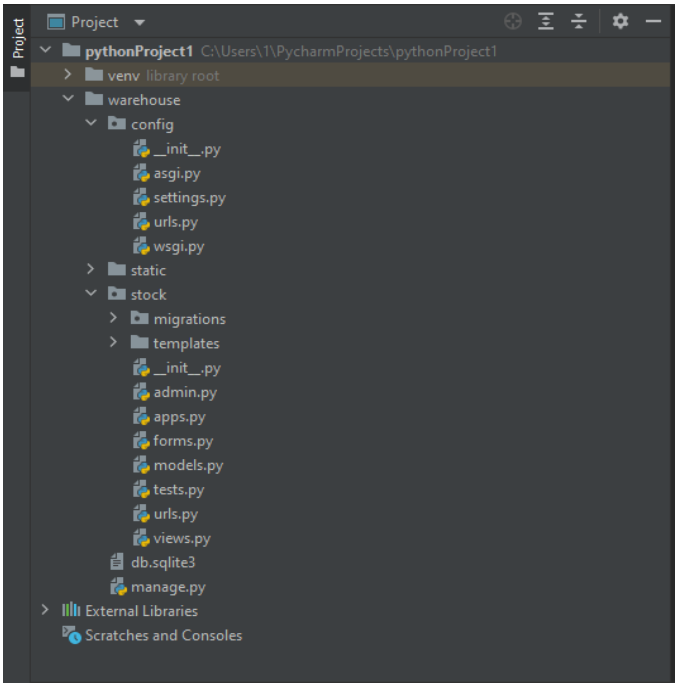
Көрініс үш тапсырманы орындайды: ол HTTP сұрауларын қабылдайды, әдістер мен сипаттар арқылы анықталған бизнес логикасын жүзеге асырады және сұрауларға жауап ретінде HTTP жауабын жібереді. Яғни, көрініс модельден деректерді алады және үлгілерге осы деректерге қол жеткізуге мүмкіндік береді немесе деректерді алдын ала өңдейді, содан кейін үлгілерге жібереді.

Django-да қуатты шаблондық қозғалтқыш және өзіндік белгілеу тілі бар. Үлгілер – деректерді көрсететін HTML файлдар. Файлдардың мазмұны статикалық немесе динамикалық болуы мүмкін. Үлгілерде биздес логикасы жоқ, сондықтан олар деректерді тек көрсетеді.



3.1-сурет – «Модель-Үлгі-Көрініс» архитектурасы

Django арқылы жазылған сайттар бір немесе бірнеше қосымшалардан тұрады. Оларды ажыратылатын және қосылатын етіп жасау ұсынылады. Django проекті келесідей құрылымға ие (3.2-сурет).



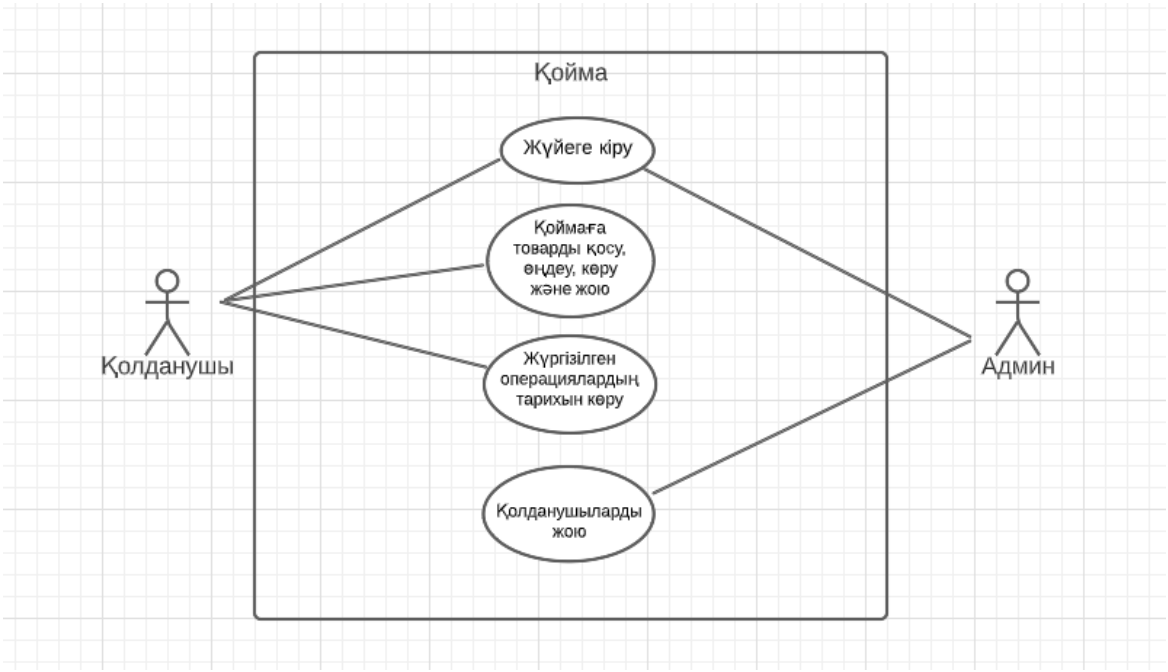
3.2-сурет – Проекттің құрылымы

Мұндағы: **“warehouse”** жобаның аты, **“stock”** веб қосымшаның логикасы орындалған қосымша. models.py – модельдер файлы, views.py – көрініс файлы және templates деректориясы ішінде архитектурада үлгі ролін атқаратын HTML файлдар сақталған.

3.2 UML диаграммалары

UML – жалпы тіл, ол UML моделі деп аталатын жүйенің дерексіз моделін жасау үшін графикалық белгілерді пайдаланатын ашық стандарт. UML негізінен бағдарламалық жүйелерді анықтау, визуализациялау, жобалау және құжаттау үшін жасалған. UML бағдарламалау тілі емес, бірақ UML үлгілері негізінде кодты генералациялау мүмкін.

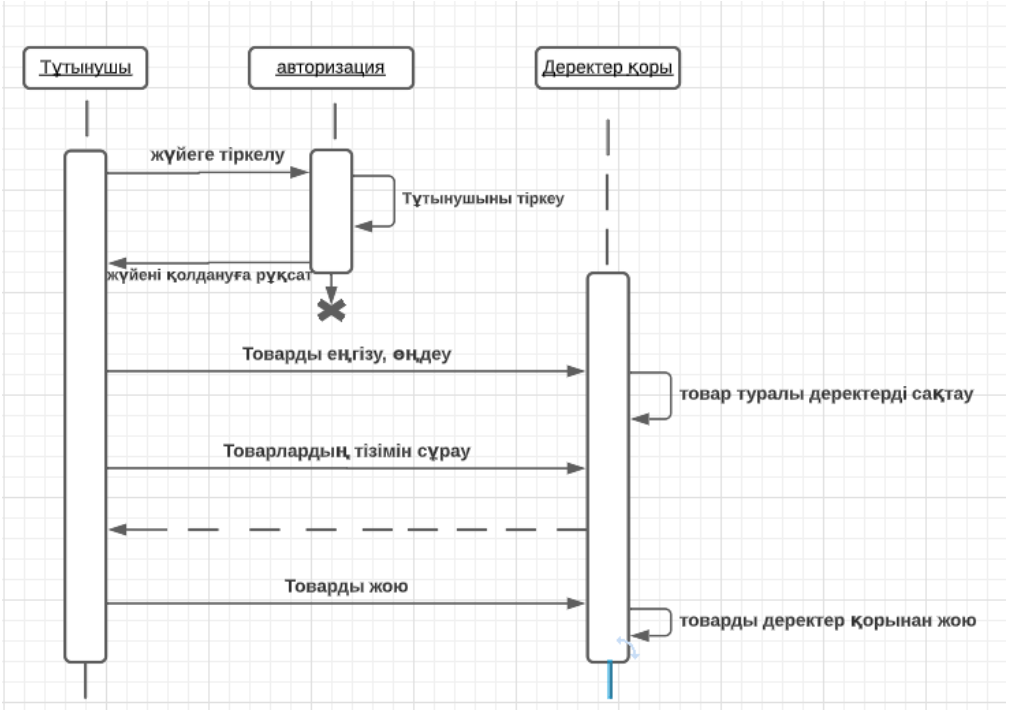
UML диаграммалардың көптеген түрлері бар. Солардың бірі прецценденттер диаграммасы. Ол тұтынушы, соңғы пайдаланушыға және әзірлеушіге жобаланған немесе бар жүйені бірге мүмкіндік беретін функционалдылық пен әрекетті сипаттау үшін қолданылады. Дипломдық жобаның прецендеттер диаграммасы (3.3-сурет).



3.3-сурет – Жобаның прецценденттер диаграммасы

Кең қолданылатын UML диаграммалардың тағы бір түрі тізбек диаграммасы. Ол бір уақыт осіндегі объектілердің белгілі бір жиынтығы үшін

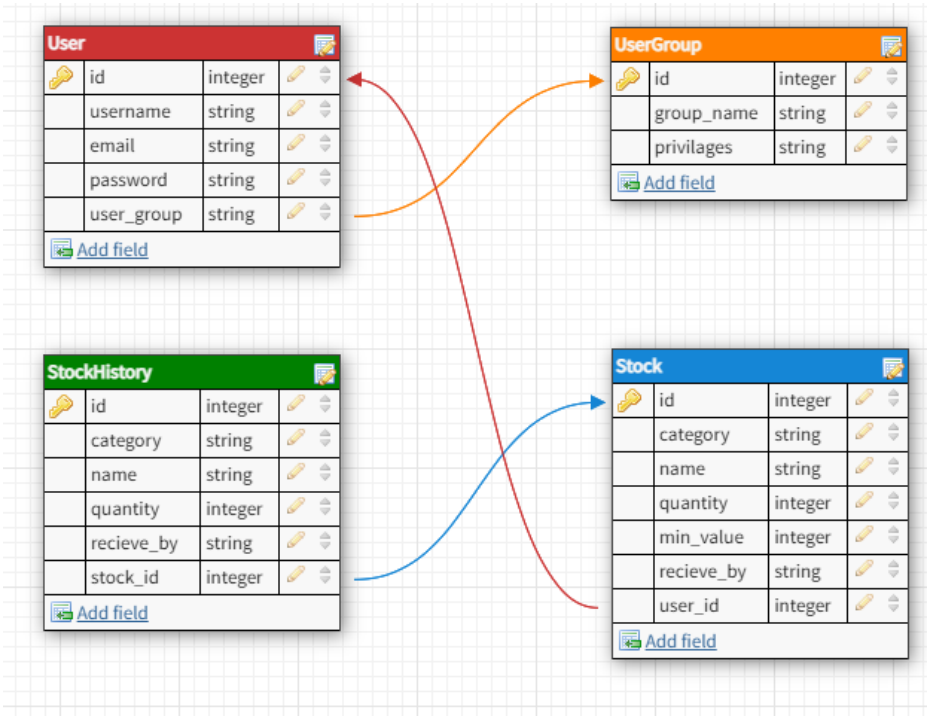
объектінің өмірлік циклін және прецедент шеңберіндегі ақпараттық жүйе субъектілерінің өзара әрекеттесуін көрсетеді. Жобаның тізбек диаграммасы келесі суретте көрсетілген (3.4-сурет).



3.4-сурет – Жобаның тізбек диаграммасы

3.3 Деректер қорының сипаттамасы

Деректер қоры – белгілі бір ережелерге сәйкес ұйымдастырылған және компьютер жадында сақталатын, белгілі бір пәндік аймақтың ағымдағы жағдайын сипаттайтын және пайдаланушылардың ақпараттық қажеттіліктерін қанағаттандыру үшін пайдаланылатын деректер жиынтығы. Пайдаланушылар туралы да, олар енгізетін тауарлар туралы да ақпарат деректер қорында сақталады. Жобаның деректер қорының негізгі кестелерінің сипаттпмасы келесідей (3.5-сурет).

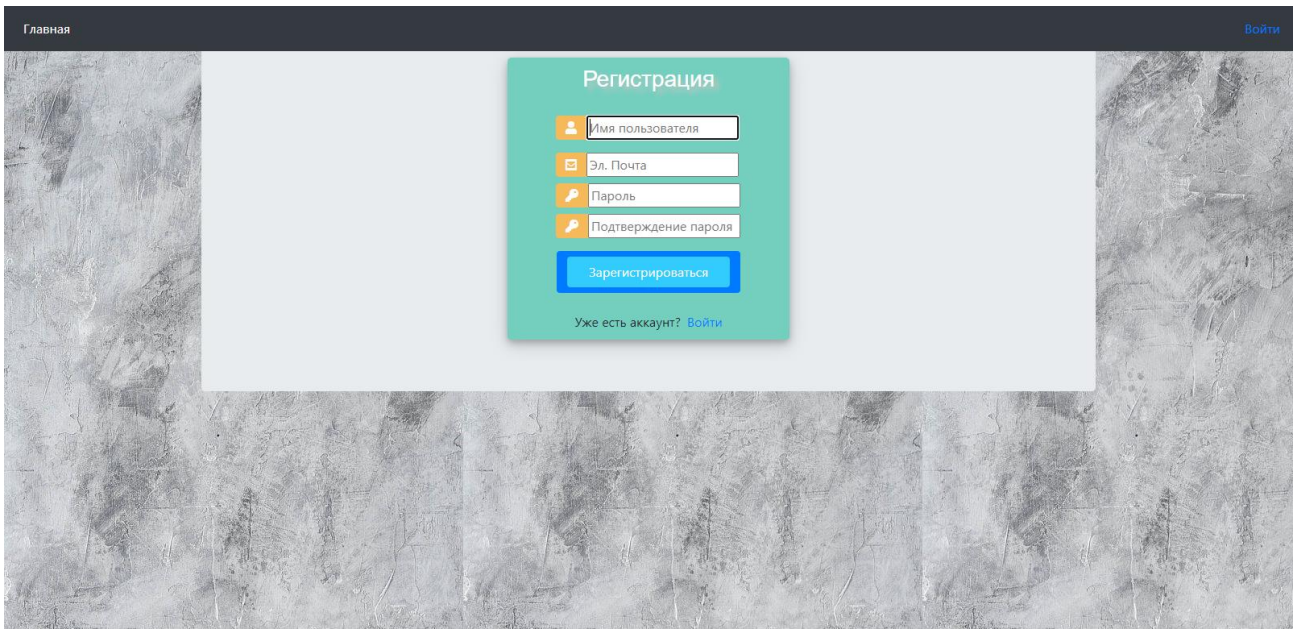


3.5-сурет – Жобаның деректер қорының сипаттамасы

3.4 Жобаның интерфейсі

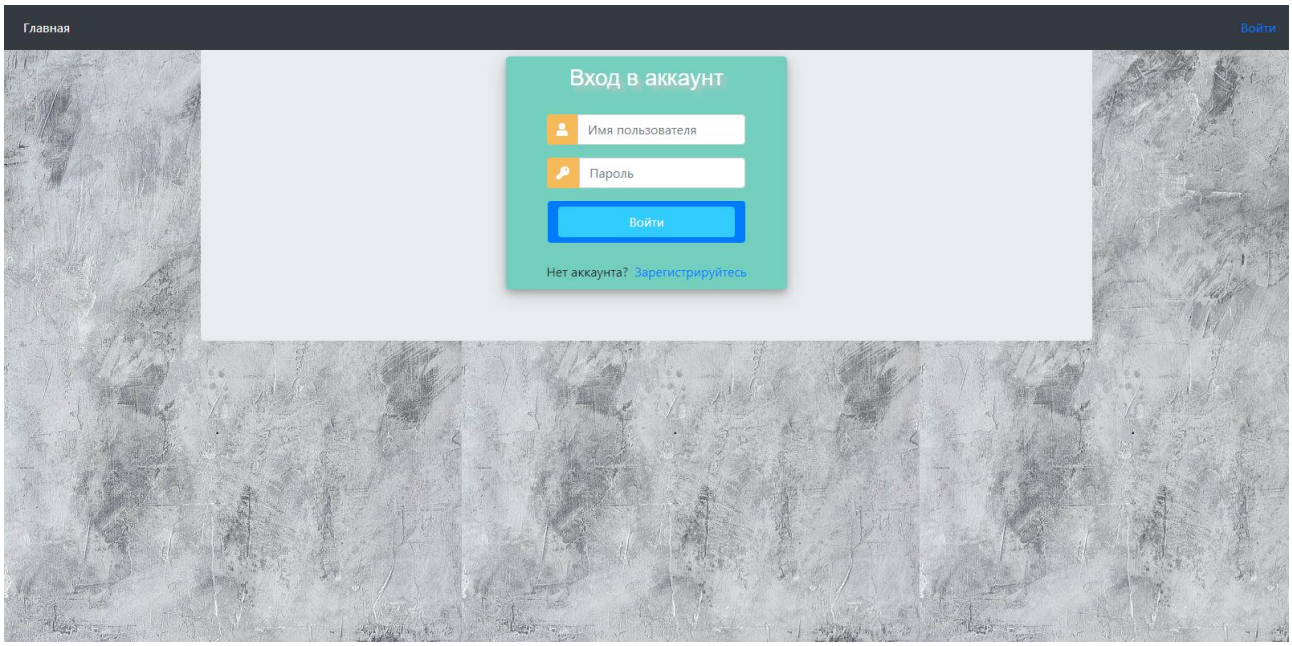
Құрылған веб-қосымшаның интерфейсінің ұқсас аналогтерінен ерекшелігі – интерфейстің басым бөлігі орталықтырылған. Бұл қолданушының жұмысын ыңғайландыра түседі.

Интерфейсті қарастырайық. Ең алдымен тіркелу парақшасы (3.6-сурет) мен кіру парақшасы (3.7-сурет).



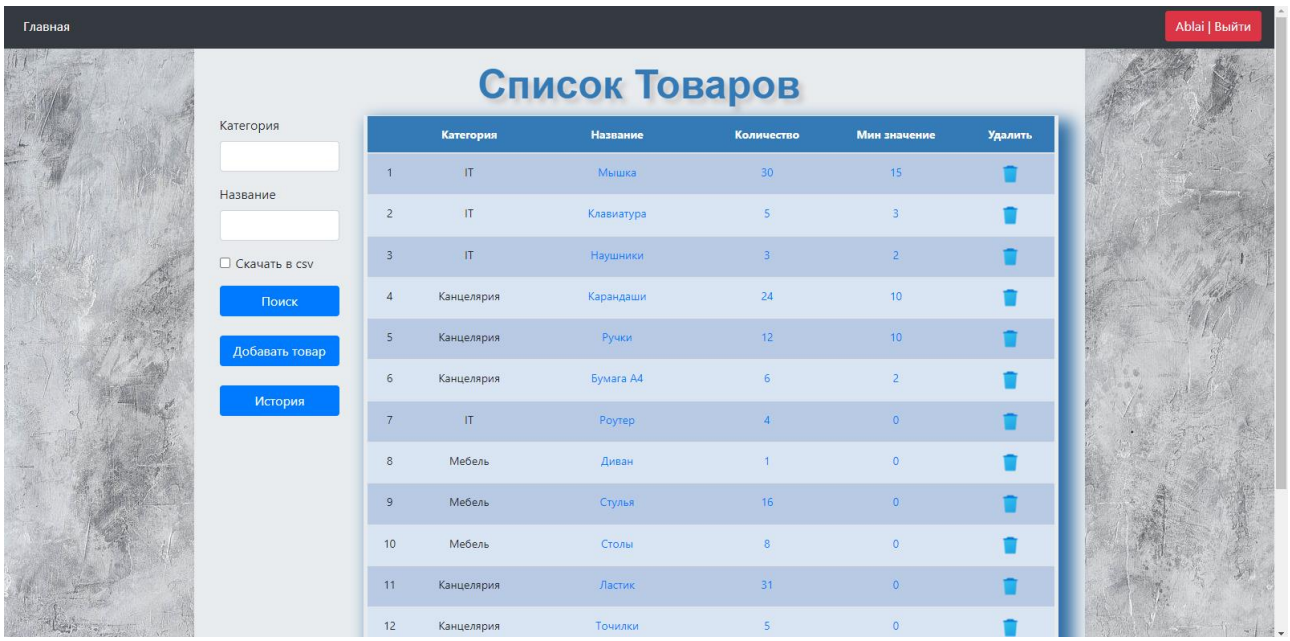
3.6-сурет – Тіркелу парақшасы

Жобада кең қолданылатын тіркелу парақшасы іске асырылған.



3.7-сурет – Кіру парақшасы

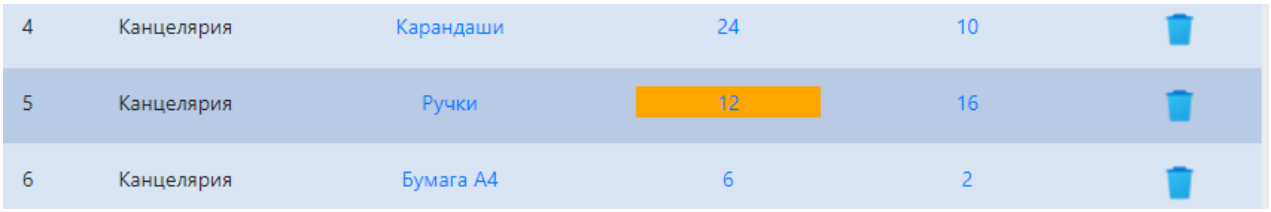
Сайтқа кірген соң қолданушы негізгі жұмыс парақшасына кіреді. Жаңадан тіркелген қолданушыда тізім бос болады, ал бұрын сайтпен жұмыс істеп тауарларды еңгізген қолданушы өзінің тізімін көреді (3.8-сурет).



3.8-сурет – «Список Товаров» (тауарлар тізімі) парақшасы

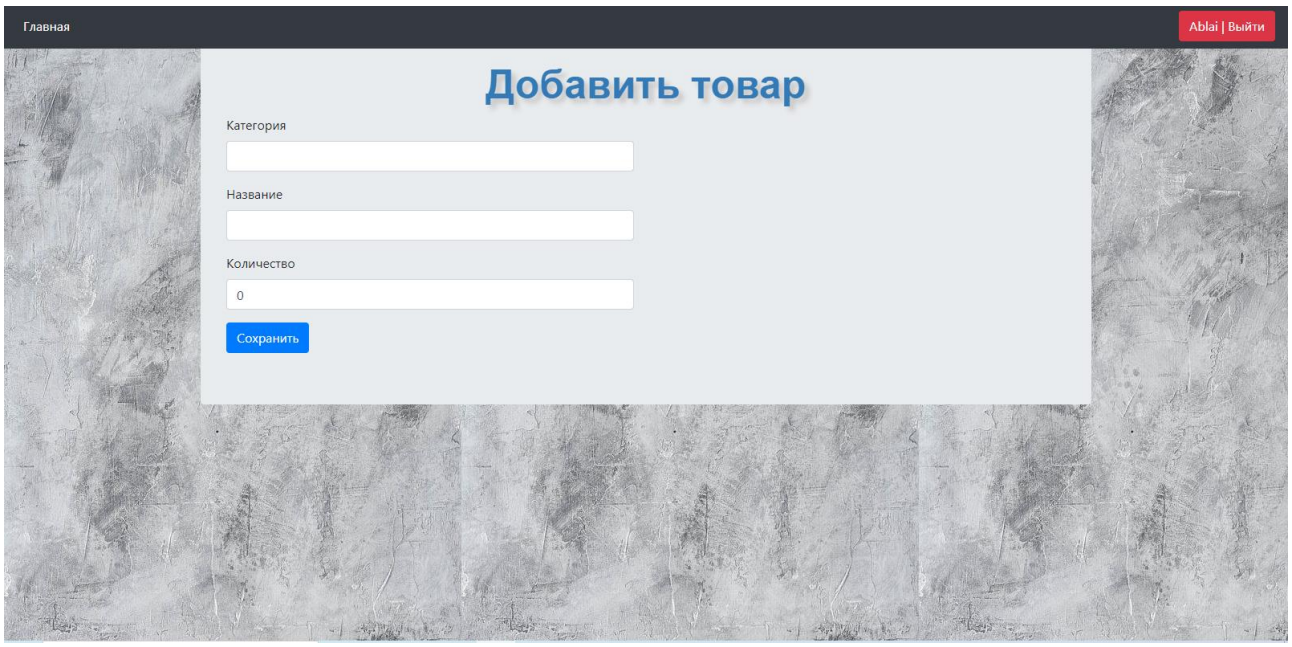
Бұл парақшада тізімнің сол жағында «Поиск», «Добавить Товар», «История» батырмалары бар. «Поиск» батырмасы жоғары толтырынған формаларға байланысты тауарлар тізімінен сәйкестерін көрсетеді. Егер «скачать в csv» қосылып тұрса онда тауардың тізімі csv форматта жүктеледі. «Добавить Товар» батырмасы тауарды қосы парақшасына жібереді. «История» тауарлармен жүзеге асырылған операциялар парақшасын көрсетеді.

Тізімге келетін болсақ бұл жерде барлық тауарлардың категориясы, атауы қоймадағы саны, минималды сан яғни қоймада тауардың саны сол саннан аз болса қолданушыға ескерту көрсетіледі (3.9-сурет). Оны қолданушы өзі қоя алады



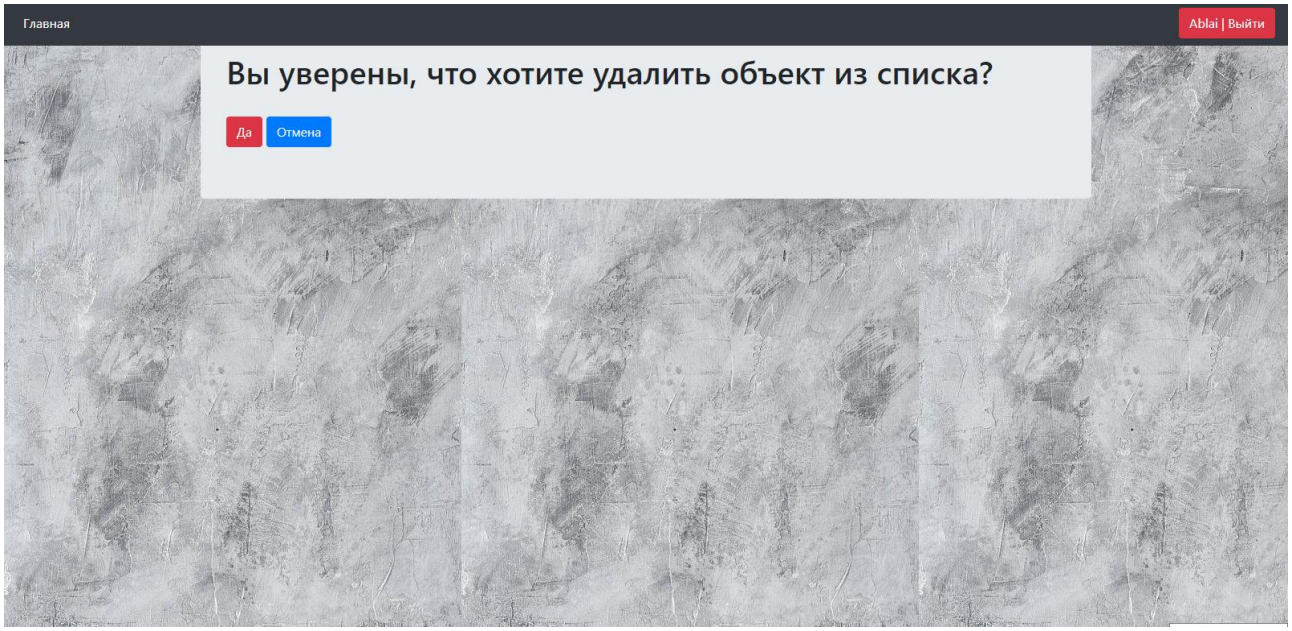
3.9-сурет – Қоймада тауар саны берілгеннен аз

Тізімге тауарды енгізу кезінде қолданушы міндетті түрде оның категориясы мен атын көрсету қажет. Сонымен қатар ол тауардың санын енгізе алады. Енгізбеген жағдайда автоматты түрде 0 қойылады (3.10-сурет).



3.10-сурет – Тауарды қосу парақшасы

Тауар тізімі парақшасы арқылы қолданущы тауарды жоя алады. Ол үшін тауардың сипаттамасы жанындағы жәшік суретіне басу керек. (3.11-сурет)

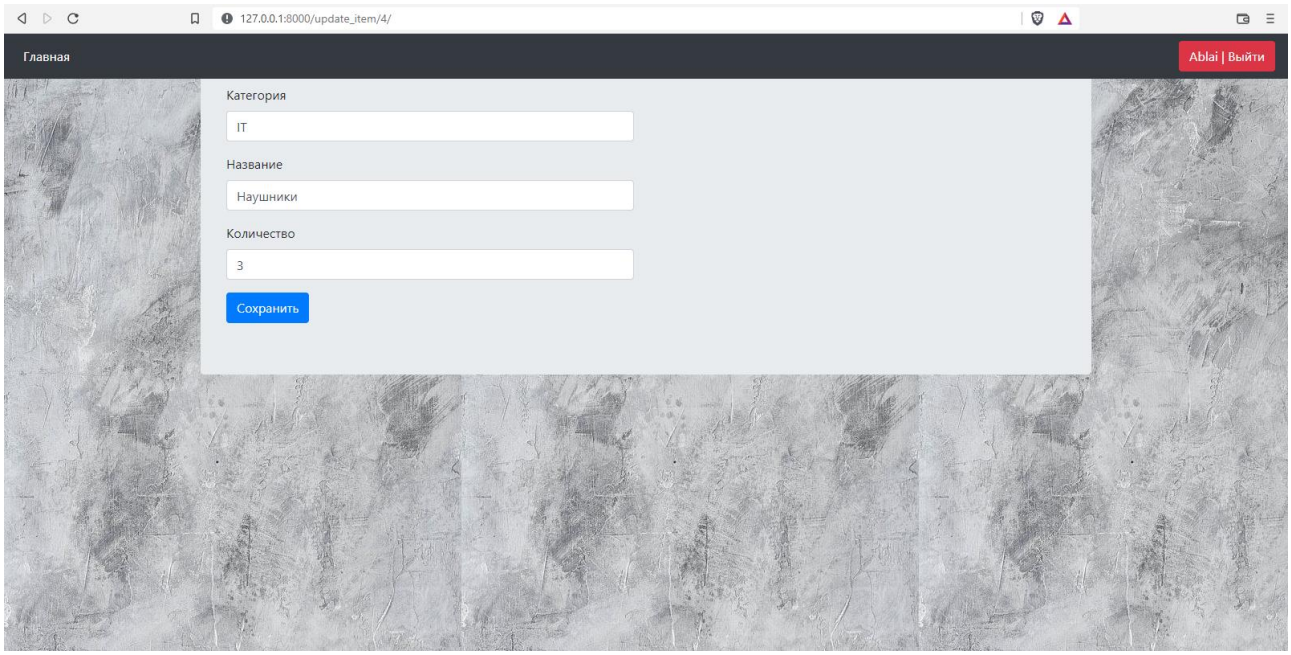


3.11-сурет – Тауарды жою парақшасы

«Да» батырмасын басқан кезде таңдалған тауар тізімнен жойылады. «Отмена» батырмасын басқан кезде жою операциясы орындалмайды.

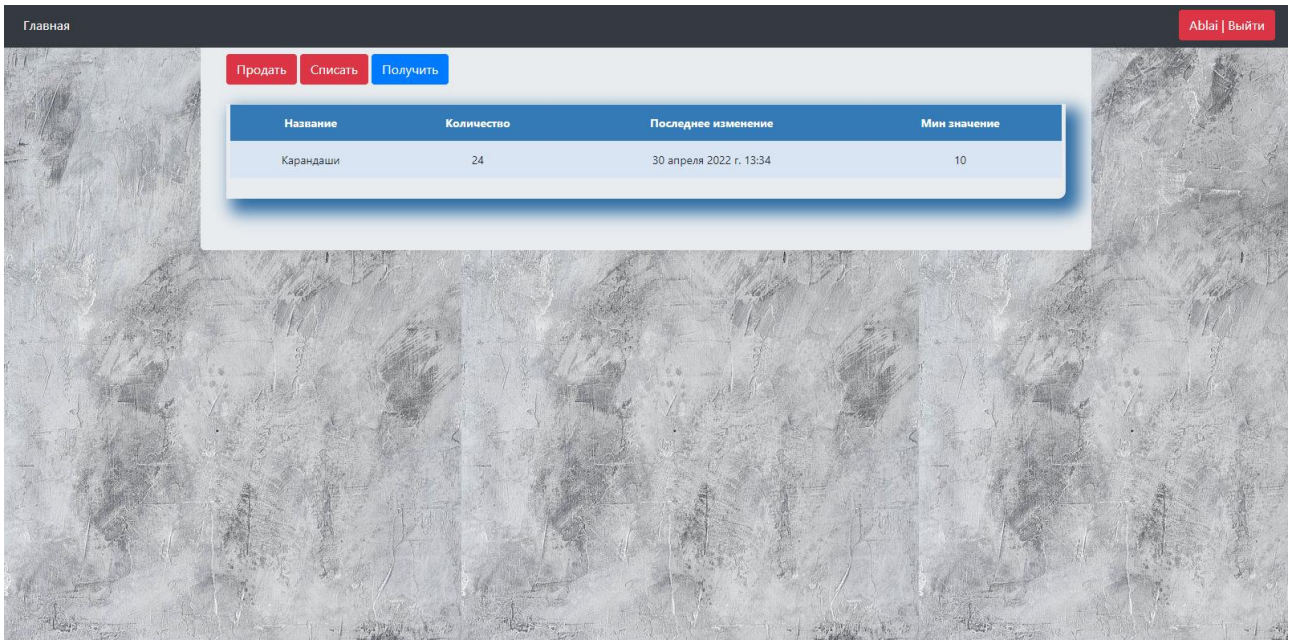
Тауар тізімі парақшасында тауардың атауына басқан кезде тауарды өзгертуге мүмкіндік беретін парақша ашылады. Ол жерде тауардың атын,

категориясын және тауар санын (тауар операциясына жатпайды) өзгертуге болады (3.12-сурет).



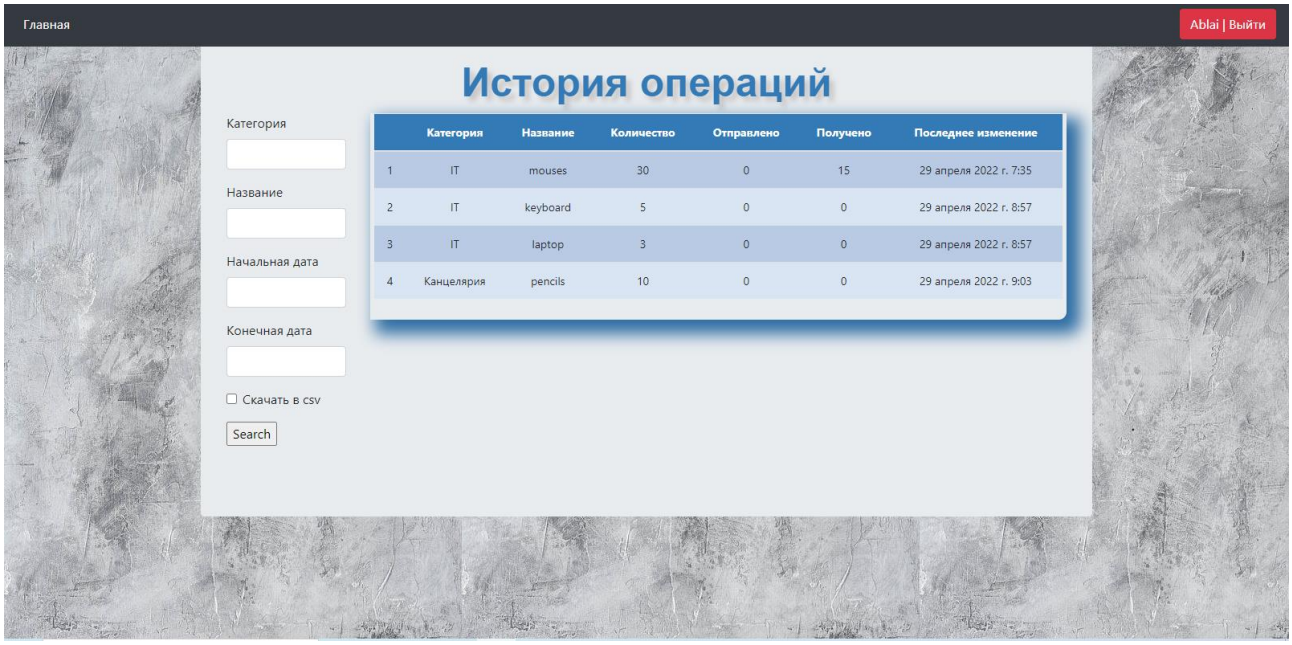
3.12-сурет – Тауарды өзгерту парақшасы

Тауар тізімі парақшасында тауардың санына басқан кезде тауарды сату, есептен шығарып тастау немесе қабылдау мүмкіндік беретін батырмалар бар парақша ашылады (3.13-сурет).



3.13-сурет – Тауармен операция жасау парақшасы

Жасалған операцияларды берілген уақытқа байланысты көруге болады. Сонымен қатар операциялар тарихы парақшасында тауардың соңғы өзгерісі жазылады. Операциялар тарихын тауар тізімі сияқты csv форматында жүктеп алуға болады (3.14-сурет).



3.14-сурет – Операциялар тарихы парақшасы

Қорытынды

Қазіргі уақытта кез келген ұйым қызмет саласына және айналымдағы тауар түріне қарамастан сақтауды дұрыс басқару мәселесімен бетпе-бет келеді. Тіпті саудамен айналыспайтын ұйымдар заттардың есебін жүргізуге міндетті болады. Сондықтан бұл бизнес процестің негізгі ережелері мен ұйымдастырылуын білу маңызды және оны елемеуге болмайды. Қоймалық есебті жүргізу көптеген кәсіпкерлер мен ұйымдарға көмектесті және қазіргі кезде де көмектесіп келеді. Сол себепті бұл істі заманауи талаптарға сай жаңалау қажет.

Дипломдық жұмыстың мақсаты қоймалық есебті жеңілдететін веб-қосымшаны әзірлеу болатын. Қолданыстағы технологияларды және тексерілген теорияны қолдана отырып қойылған мақсатқа қол жеткізілген болатын. Қосымша тауарды сақтау және сатып алу, қабылдау процесстерін қолдайды және жүргізілген операциялар тарихын сақтайды. Әзірленген қосымша қолданысқа жеңіл, бұрын ондай іспен айналыспаған қолданушыларға тез бейімделуге мүмкіндік береді.

Пайдаланылған әдебиеттер тізімі

1 Джанго документациясы // Сайттың электрондық нұсқасы https://docs.djangoproject.com/en/4.0/

2 Есеп пен аудиттің тарихы // Сайттың электрондық нұсқасы https://studbooks.net/1516811/buhgalterskiy_uchet_i_audit/istoriya_buhgalterskogo_ucheta_i_audita

3 MySQL документациясы // Сайттың электрондық нұсқасы https://dev.mysql.com/doc/

4 Ақпараттық технологиялардың қойма қызметін ұйымдастырудағы рөлі // Сайттың электрондық нұсқасы https://cyberleninka.ru/article/n/rol-informatsionnyh-tehnologiy-v-organizatsii-skladskoy-deyatelnosti

5 UML диаграммалар // Сайттың электрондық нұсқасы https://evergreens.com.ua/ru/articles/uml-diagrams.html

6 Джанго жобаның архитектурасы // Сайттың электрондық нұсқасы https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Introduction

7 PyCharm IDE // Сайттың электрондық нұсқасы https://ru.wikipedia.org/wiki/PyCharm

8 Реляциялық деректер қоры // Сайттың электрондық нұсқасы https://www.oracle.com/cis/database/what-is-a-relational-database/

9 UML диаграммаларын қолданудын маңызы // Сайттың электрондық нұсқасы https://habr.com/ru/post/458680/

10 MySQL Workbench бағдарламасын қосу // Сайттың электрондық нұсқасы https://selectel.ru/blog/tutorials/mysql-workbench-installation/

11 Форсье Джефф, Биссекс Пол Django. Разработка веб-приложений на Python.

12 Қойма түрлері мен пайда болу тарихы // Сайттың электрондық нұсқасы https://en.wikipedia.org/wiki/Warehouse

13 Python документациясы // Сайттың электрондық нұсқасы https://www.python.org/doc/

14 Дерек қорды құру // Сайттың электрондық нұсқасы https://habr.com/ru/post/193136/

15 Гимельштейн Е.А., Годван Д.Ф., Левченко К.О., RFID системы для автоматизации складского учета // Сайттың электрондық нұсқасы https://cyberleninka.ru/article/n/rfid-sistemy-dlya-avtomatizatsii-skladskogo-ucheta/viewer

А қосымшасы
(міндетті)

Техникалық тапсырма

А.1.5 Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін сайтын құру үшін техникалық тапсырма.

Бұл техникалық тапсырма қоймалық процесстерін қолдайтын жүйені әзірлеуге қолданылады. Бұл жүйе саудамен айналысатын да айналыспайтын ла тұлғалар қолданады.

А.1.5.1 Мақсаты

Жүйе қолданущыларға қоймалық процесстерді жеңілдету мақсатымен жасалған.

А.1.5.3 Функционалды сипаттамаларға қойылатын талаптар

Жүйе келесі функциларды орындауы қажет:
- жүйеге қолданушаларды тіркеу;
- тауарлар туралы ақпаратты сақтау;
- тауарларды тізімге енгізу, тізімнен жою, өңдеу;
- тауарлармен жүргізілген операцияларды сақтау;
- тауар тізімі мен операциялар тарихын жүктеу;

А.1.5.4 Сенімділікке қойылатын талаптар

Сақталған ақпараттың тұтастығын сақтау. Қолданушылар туралы деректерінің құпиялылығын қамтамасыз ету.

А қосымшаның жалғасы

А.1.5.5 Ақпараттық және бағдарламалық үйлесімдікке қойылатын талаптар

Жүйе барлық IBM-үйлесімді компьютерлер мен браузерге кіре алатын смартфондарда қосылуы керек .Сонымен қатар жүйе барлық браузерлерде жұмыс істеуі тиіс.

Б қосымшасы
(міндетті)

Бағдарлама мәтіні

Жобаның «Модель-Көрініс-Үлгі» архитектурасына байланысты негізгі код бөлімдері.

1 Модель

```
from django.db import models
from django.contrib.auth.models import User
from django.conf import settings


type_choices = [("шт", "шт"), ("кг", "кг"), ("л", "л"), ("гр", "гр"), ("м3", "м3"), ("м2", "м2")]


class CountType(models.Model):
    count_type = models.CharField(max_length=20, choices=type_choices)

    def __str__(self):
        return self.count_type


class Stock(models.Model):
    category = models.CharField(max_length=50, blank=True, default='')
    item_name = models.CharField(max_length=100, blank=True, null=True)
    count_type = models.ForeignKey(CountType, on_delete=models.CASCADE)
    cost = models.IntegerField(default='0', blank=True, null=True)
    quantity = models.PositiveIntegerField(default='0', blank=True, null=True)
    receive_quantity = models.IntegerField(default='0', blank=True, null=True)
    receive_by = models.CharField(max_length=50, blank=True, null=True)
    issue_quantity = models.IntegerField(default='0', blank=True, null=True)
    issue_by = models.CharField(max_length=50, blank=True, null=True)
    issue_to = models.CharField(max_length=50, blank=True, null=True)
    created_by = models.CharField(max_length=50, blank=True, null=True)
    reorder_level = models.IntegerField(default='0', blank=True, null=True)
```

```
    last_updated = models.DateTimeField(auto_now_add=False, auto_now=True)
    timestamp = models.DateTimeField(auto_now_add=True, auto_now=False)

    def __str__(self):
        return self.item_name


class StockHistory(models.Model):
    category = models.CharField(max_length=100, blank=True, default='')
    item_name = models.CharField(max_length=100, blank=True, null=True)
    quantity = models.PositiveIntegerField(default='0', blank=True, null=True)
    cost = models.IntegerField(default='0', blank=True, null=True)
    count_type = models.ForeignKey(CountType, on_delete=models.CASCADE)
    receive_quantity = models.IntegerField(default='0', blank=True, null=True)
    receive_by = models.CharField(max_length=50, blank=True, null=True)
    issue_quantity = models.IntegerField(default='0', blank=True, null=True)
    issue_by = models.CharField(max_length=50, blank=True, null=True)
    issue_to = models.CharField(max_length=50, blank=True, null=True)
    created_by = models.CharField(max_length=50, blank=True, null=True)
    reorder_level = models.IntegerField(default='0', blank=True, null=True)
    last_updated = models.DateTimeField(auto_now_add=False, auto_now=False,
null=True)
    timestamp = models.DateTimeField(auto_now_add=False, auto_now=False,
null=True)
```

2 Көрініс

```
from django.contrib.auth import authenticate, login
from django.shortcuts import render, redirect
from .models import Stock, StockHistory
from .forms import *
from django.contrib.auth.decorators import login_required
from django.contrib.auth.forms import UserCreationForm
from django.http import HttpResponse
from django.contrib import messages
import csv
```

Б қосымшасының жалғасы

```
def registerPage(request):
    form = RegisterUserForm()
    if request.method == 'POST':
        form = RegisterUserForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('login')
    context = {'form': form}
    return render(request, "user/register.html", context)


def loginPage(request):
    context = {}
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(request, username=username, password=password)

        if user is not None:
            login(request, user)
            return redirect('list_item')

    return render(request, "user/login.html", context=context)


def home(request):
    title = 'Home page'
    context = {
        "title": title,
    }
    return redirect('home.html')


@login_required
```

Б қосымшасының жалғасы

```
def list_item(request):
    form = StockSearchForm(request.POST or None)
    title = 'Список Товаров'
    queryset = Stock.objects.all()
    context = {
        "form": form,
        "title": title,
        "queryset": queryset,
    }
    if request.method == 'POST':
        category = form['category'].value()
        queryset = Stock.objects.filter(
            item_name__icontains=form['item_name'].value()
        )

        if category != '':
            queryset = Stock.objects.filter(
                category__icontains=form['category'].value()
            )
            #queryset = queryset.filter(category=category)

        if form['export_to_CSV'].value():
            response = HttpResponse(content_type='text/csv')
            response['Content-Disposition'] = 'attachment; filename="Список
товаров.csv"'
            writer = csv.writer(response)
            writer.writerow(['КАТЕГОРИЯ', 'НАЗВАНИЕ', 'КОЛИЧЕСТВО'])
            instance = queryset
            for stock in instance:
                writer.writerow([stock.category, stock.item_name, stock.quantity])
            return response
        context = {
            "form": form,
            "title": title,
            "queryset": queryset,
        }
```

```
    return render(request, "list_item.html", context)


@login_required
def add_items(request):
    form = StockCreateForm(request.POST or None)
    if form.is_valid():
        form.save()
        return redirect('/list_item')
    context = {
        "form": form,
        "title": "Добавить товар",
    }
    return render(request, "add_item.html", context)


def update_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = StockUpdateForm(instance=queryset)
    if request.method == 'POST':
        form = StockUpdateForm(request.POST, instance=queryset)
        if form.is_valid():
            form.save()
            return redirect('/list_item')

    context = {
        'form': form
    }
    return render(request, 'add_item.html', context)


def delete_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    if request.method == 'POST':
        queryset.delete()
        return redirect('/list_item')
```

```
    return render(request, 'delete_item.html')


def stock_detail(request, pk):
    queryset = Stock.objects.get(id=pk)
    context = {
        "queryset": queryset,
    }
    return render(request, "stock_detail.html", context)


def issue_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = IssueForm(request.POST or None, instance=queryset)
    if form.is_valid():
        instance = form.save(commit=False)
        instance.receive_quantity = 0
        instance.quantity -= instance.issue_quantity
        instance.issue_by = str(request.user)
        instance.save()
        return redirect('/stock_detail/'+str(instance.id))
    context = {
        "title": 'Продажа ' + str(queryset.item_name),
        "queryset": queryset,
        "form": form,
        "username": 'Покупатель: ' + str(request.user),
    }
    return render(request, "add_item.html", context)


def remove_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = RemoveForm(request.POST or None, instance=queryset)
    if form.is_valid():
        instance = form.save(commit=False)
        instance.quantity -= instance.issue_quantity
```

```
        instance.save()
        return redirect('/stock_detail/'+str(instance.id))
    context = {
        "title": 'Списание ' + str(queryset.item_name),
        "queryset": queryset,
        "form": form,
    }
    return render(request, "add_item.html", context)


def receive_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = ReceiveForm(request.POST or None, instance=queryset)
    if form.is_valid():
        instance = form.save(commit=False)
        instance.issue_quantity = 0
        instance.quantity += instance.receive_quantity
        instance.receive_by = str(request.user)
        instance.save()

        return redirect('/stock_detail/' + str(instance.id))
    context = {
        "title": 'Закупка ' + str(queryset.item_name),
        "queryset": queryset,
        "form": form,
        "username": 'От кого: ' + str(request.user),
    }
    return render(request, "add_item.html", context)


def reorder_level(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = ReorderLevelForm(request.POST or None, instance=queryset)
    if form.is_valid():
        instance = form.save(commit=False)
        instance.save()
```

```
        return redirect("/list_item")
    context = {
            "instance": queryset,
            "form": form,
        }
    return render(request, "add_item.html", context)


@login_required
def list_history(request):
    title = 'История операций'
    queryset = StockHistory.objects.all()
    form = StockHistorySearchForm(request.POST or None)
    context = {
        "title": title,
        "queryset": queryset,
        "form": form
    }
    if request.method == 'POST':
        category = form['category'].value()
        queryset = StockHistory.objects.filter(
            item_name__icontains=form['item_name'].value(),
            last_updated__range=[
                form['start_date'].value(),
                form['end_date'].value()
            ]
        )

        if category != '':
            queryset = queryset.filter(category_id=category)

        if form['export_to_CSV'].value():
            response = HttpResponse(content_type='text/csv')
            response['Content-Disposition'] = 'attachment; filename="История
операций.csv"'
            writer = csv.writer(response)
```

Б қосымшасының жалғасы

```
        writer.writerow(
            ['CATEGORY',
             'ITEM NAME',
             'QUANTITY',
             'ISSUE QUANTITY',
             'RECEIVE QUANTITY',
             'RECEIVE BY',
             'ISSUE BY',
             'LAST UPDATED'])
        instance = queryset
        for stock in instance:
            writer.writerow(
                [stock.category,
                 stock.item_name,
                 stock.quantity,
                 stock.issue_quantity,
                 stock.receive_quantity,
                 stock.receive_by,
                 stock.issue_by,
                 stock.last_updated])
        return response
    context = {
        "form": form,
        "title": title,
        "queryset": queryset,
    }
return render(request, "list_history.html", context)
```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТІРЛІГІ
СӘТБАЕВ УНИВЕРСИТЕТІ

Қ.И. Сәтбаев атындағы Қазақ Ұлттық Техникалық зерттеу университеті

Мамандығы 5В070400 – Есептеу техникасы және программалық қамтамасыз ету

Студент Жортулов Аблай

Тақырыбы: Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ

Дипломдық жобасын жасаушы Жортулов Аблайға қоймалық процесстерді қолдайтын веб-қосымшаны құру, жүргізілген операцияларды сақтау жүйесін енгізу міндеттері қойылған

Жортулов Аблай Python бағдарламалау тілін, бұл бағдарламалау тіліне арналған Django фреймворкын, MySQL технологияларын жақсы меңгере отырып, админ бетін пайдалану, веб-қосымшаны құру жолдарын жүзеге асырды. Бағдарламаны құру туралы теориялық ақпарат келтірілген. Нәтижені әрі қарай өңдеу және жақсарту кеңес етілді.

Дипломдық жобаның негізгі мазмұнын құрайтын барлық нәтижелер студенттің өз бетімен алынған, ал жетекші мәселенің қойылуы, кеңестер мен *нәтижелерді талдау ғана тиісті.*

Қорытындылай келе, дипломдық жоба 5В070400 – «Есептеу техникасы және прогрпммалық қамтамасыз ету» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және 5В070400 – «Есептеу техникасы және прогрпммалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады және дипломдық жобаны қорғауға жіберіледі деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ _____ **Н.Э. Алпысбай**
«Программалық инженерия»
Кафедрасының лекторы «10» 05 2022 жыл

Университеттің жүйе администраторы мен Академиялық мәселелер департаменті директорының ұқсастық есебіне талдау хаттамасы

Жүйе администраторы мен Академиялық мәселелер департаментінің директоры көрсетілген еңбекке қатысты дайындалған Плагиаттың алдын алу және анықтау жүйесінің толық ұқсастық есебімен танысқанын мәлімдейді:

Автор: Жортулов Аблай Сәкенұлы

Тақырыбы: Қоймада тауарларды сатып алу және сақтау процестерін қолдауды қамтамасыз ету үшін ақпараттық жүйені жобалау және әзірлеу

Жетекшісі: Жибек Алибиева

1-ұқсастық коэффициенті (30): 9.8

2-ұқсастық коэффициенті (5): 2.4

Дәйексөз (35): 5.5

Әріптерді ауыстыру: 1

Аралықтар: 0

Шағын кеңістіктер: 0

Ақ белгілер: 5

Ұқсастық есебін талдай отырып, Жүйе администраторы мен Академиялық мәселелер департаментінің директоры келесі шешімдерді мәлімдейді :

☐ Ғылыми еңбекте табылған ұқсастықтар плагиат болып есептелмейді. Осыған байланысты жұмыс өз бетінше жазылған болып санала отырып, қорғауға жіберіледі.

☐ Осы жұмыстағы ұқсастықтар плагиат болып есептелмейді, бірақ олардың шамадан тыс көптігі еңбектің құндылығына және автордың ғылыми жұмысты өзі жазғанына қатысты күмән тудырады. Осыған байланысты ұқсастықтарды шектеу мақсатында жұмыс қайта өңдеуге жіберілсін.

☐ Еңбекте анықталған ұқсастықтар жосықсыз және плагиаттың белгілері болып саналады немесе мәтіндері қасақана бұрмаланып плагиат белгілері жасырылған. Осыған байланысты жұмыс қорғауға жіберілмейді.

Негіздеме:

Күні 13.05.2020

Кафедра меңгерушісі