

**МИНИСТЕРСТВО РАЗВИТИЯ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛЬ-ХОРЕЗМИЙ**

ФАКУЛЬТЕТ «КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ»

Для студентов направления бакалавриата
60610500 – «Компьютерный инжиниринг» («ИТ-сервис»)

МЕТОДИЧЕСКОЙ УКАЗАНИЯ
к выполнению проектной работы №1

Ташкент - 2022

Методические указания составлены по типовому учебному плану, утвержденному Министерством Высшего и Среднего Специального образования Республики Узбекистан.

Составители:

Маликова М.Т. – ТУИТ имени Мухаммада аль-Хоразмий доцент
кафедры «Информационные технологии»

Джураев Т.Б. – ТУИТ имени Мухаммада аль-Хоразмий, старший
преподаватель кафедры «Информационные
технологии».

Рецензенты:

Мирзаев Д.А. - к.т.н. доцент, заведующий кафедрой «
Информационных систем и информационных технологий»
Ташкентский финансовый институт

Мансурова М.Я. - ТУИТ имени Мухаммада аль-Хоразмий, к.т.н. (PhD)
доцент кафедры «Информационные технологии»

Методическое указание обсуждена и утверждена на Учебно-методическом совете факультета «Компьютерный инжиниринг» и рекомендована для печати.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ		4
ГЛАВА 1.	ПРОЕКТНАЯ РАБОТА И ТРЕБОВАНИЯ К НЕЙ.....	6
1.1.	Основные понятия о проектной работе.....	6
1.2.	Цели и задачи проектной работы.....	6
1.3.	Порядок выбора темы для выполнения проектной работы.....	7
1.4.	Предлагаемые темы для выполнения проекта	7
ГЛАВА 2.	ОРГАНИЗАЦИОННЫЕ АСПЕКТЫ ПРОЕКТНОЙ РАБОТЫ.....	12
2.1.	Организационная работа при выполнении проектной работы..	12
2.2.	Пояснительная записка проектной работы	12
2.3.	Структура проектной работы и ее основные части	17
2.4.	Порядок написания введения и основной части	18
ГЛАВА 3.	ПРАВИЛА ПОДГОТОВКИ ПРОЕКТНЫХ РАБОТ И ТРЕБОВАНИЯ К НИМ.....	21
3.1.	Требования к чертежам	22
3.2.	Требования к вводу документов, связанных с программами....	22
3.3.	Порядок составления списка используемой литературы	23
ГЛАВА 4.	ПОРЯДОК ЗАЩИТЫ ПРОЕКТНОЙ РАБОТЫ	24
4.1.	Подготовка к защите проектной работы.....	24
4.2.	Порядок работы комиссии, принимающей проектные работы..	24
4.3.	Защита проектной работы	24
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....		26
Приложение 1.....		28
Приложение 2.....		29
Приложение 3.....		30
Образец для выполнения проектной работы.....		31

ВВЕДЕНИЕ

На сегодняшний день одним из основных направлений современных информационных технологий является инжиниринг услуг информационных технологий, которое в основном заключается в решении задач пользователей.

В результате стремительного развития информационных и коммуникационных технологий появляются возможности их использования в различных сферах. Если раньше областью применения информационных систем была только автоматизация бухгалтерского учета, то теперь они используются в различных отраслях народного хозяйства.

Результатом применения информационных систем организации является максимальное использование информационных систем, снижение управленческих ошибок при ведении современной хозяйственной деятельности и обеспечение того, чтобы предприятие всегда было под контролем.

Поэтому в условиях конкуренции только правильное проектирование систем с помощью современных информационных технологий увеличивает вероятность успеха использующих их предприятий.

Цель проектной работы. Заключается в изучении всеми студентами, обучающихся по направлению Компьютерная инженерия: «ИТ-сервис» следующих аспектов: инженерия услуг информационных технологий и управление услугами, информационные системы и их компоненты, организация и создание систем, классификация информационных систем по структуре и факторам, конкретных особенностей и возможностей проектируемой системы, а также основы применения информационных систем в различных сферах человеческой деятельности. Основное внимание проектной работы направлено на использование информационных систем в сфере телекоммуникаций и информационных технологий.

Задача проектной работы - заключается в ознакомлении студентов с видами ИТ-услуг, технической структурой и программным обеспечением

информационных систем и производство этих средств, проведении научных исследований, создании новых видов услуг. Проектная работа задает в качестве задания студентам, обучающиеся по направлению Компьютерная инженерия: «ИТ-сервис» и изучающих в рамках каждой специальности и освоивших основные понятия с теоретической и практической точки зрения.

Проектная работа состоит из 3-х частей: анализ выбранной предметной области и постановка задачи, вычислительное или программное обеспечение и аналитическая часть, сдаётся путём защиты проектной работы.

ГЛАВА 1. ПРОЕКТНАЯ РАБОТА И ТРЕБОВАНИЯ К НЕЙ

1.1. Основные понятия о проектной работе

В ходе выполнения проекта студент изучает специальные науки, имеющийся новейший опыт, научно-техническую литературу и самостоятельно выдает результат, отражающий современное решение по его специальности. Темы проектных работ задаются преподавателем студентам, полностью освоившим предмет и владеющими теоретическими и практическими знаниями по предмету. Руководство проектной работой определяется указанием общего направления темы и осуществляется посредством указаний и рекомендаций.

Проектная работа должна соответствовать всем предъявляемым к ней требованиям и должна быть завершена в сроки, отведенные на ее выполнение.

1.2. Цели и задачи проектной работы

Цель проектной работы:

- укрепление знаний учащихся по данному предмету и смежным дисциплинам, расширение диапазона теоретических и практических знаний;
обучение студентов в современный период научно-технического развития к самостоятельному более глубокому изучению инженерно-исследовательских вопросов, решению их в совершенстве во всех аспектах;
- обучение к формированию навыков выполнения расчетно-чертежных работ с использованием научно-технической литературы, готовых программ, вычислительных технических средств, средств автоматизации.

Задача проектной работы – научить студентов использовать современные методы и компьютеры при решении актуальных задач в области ИКТ, решении конкретной задачи по проектированию.

1.3. Порядок выбора темы для выполнения проектной работы

Согласно характеристикам профессиональных навыков, у студентов, обучающихся по направлению «ИТ-сервис», требуются знания по нескольким дисциплинам. Для освоения дисциплин требуются теоретические знания и умение работать с прикладным программным обеспечением. В этой связи, при выборе тем для проектной работы необходимо проанализировать темы в теоретических и практических аспектах, необходимые для народного хозяйства, экономики и социальной сферы, должны быть направлены на проектирование или создание некоторых частей отраслевых систем, нуждающиеся в усовершенствовании оборудования, средств и программного обеспечения.

Темы для проектной работы задаются отдельно для каждого студента.

В некоторых случаях тема, посвященная сложной проблеме, может быть отдана 2 студентам. При этом, общая тема считается единой, и некоторые вопросы этой проблемы задаются каждому студенту в качестве задания.

Темы рассматриваются и утверждаются на заседании кафедры. Данные темы пересматриваются каждый учебный год и направляются на решение актуальных задач. Такой подход повышает значимость проекта, его научный или практический уровень.

После выбора темы проектной работы студент должен выбрать исходную гипотезу, относящуюся к теме, и на основе этой гипотезы собрать данные, а на основании собранных данных с помощью метода анализа будут представлены инструменты, программы или рекомендации.

1.4. Предлагаемые темы для выполнения проекта

1. Автоматизация получения информации о лекарственных средствах, имеющих в аптеках.
2. Автоматизация организации очередей в паспортной системе.
3. Создание онлайн-информационной системы железнодорожных билетов и рейсов.

4. Организация онлайн-покупки авиабилетов.
5. Проектирование локальной сети профессионального колледжа.
6. Организация онлайн-заявки в санатории.
7. Получение информации о коммунальных платежах с помощью информационных технологий.
8. Создание кредитных калькуляторов для банковской отрасли.
9. Система онлайн-заказов с использованием информационно-коммуникационных технологий.
10. Автоматизирование вызова такси с использованием информационно-коммуникационных технологий.
11. Создание информационной системы для получения информации об адресах, телефонах и электронных адресах государственных организаций.
12. Проектирование и разработка базы данных сельского врачебного пункта.
13. Проектирование и разработка базы данных магазина женской одежды (на примере конкретного магазина).
14. Проектирование и разработка базы данных школы-интерната для детей с ограниченными возможностями.
15. Проектирование и разработка базы данных отдела страховой компании (на примере конкретной компании).
16. Проектирование и разработка базы данных интернет-магазина одежды (на примере конкретного магазина).
17. Проектирование и разработка базы данных строительной компании (на примере конкретной компании).
18. Проектирование и разработка базы данных магазина стройматериалов (на примере конкретного магазина).
19. Проектирование и разработка базы данных автозапчастей (на примере конкретного магазина).
20. Проектирование и разработка базы данных частной школы (на примере конкретной школы).

21. Проектирование и разработка базы данных библиотеки ТАТУ.
22. Проектирование и разработка базы данных магазина телефонов (на примере конкретного магазина).
23. Налаживание онлайн-обращений в банковскую систему.
24. Организация онлайн-обращений в налоговую систему.
25. Создание информационной системы для получения оперативной информации об авиарейсах.
26. Организация рекламной деятельности по недвижимости, выставленной на продажу, с использованием информационно-коммуникационных технологий.
27. Создание информационной системы по вакансиям.
28. Организация онлайн-покупки билетов на концерты и в театры.
29. Создание информационной системы электронной подписки на газеты и журналы.
30. Автоматизирование учебного процесса в профессиональных колледжах.
31. Внедрение онлайн-оплаты коммунальных услуг.
32. Создание электронного магазина по продаже компьютерной техники.
33. Создание информационной системы «Бизнескумак» в поддержку предпринимателей.
34. Проектирование и разработка базы данных предприятия по производству фарфоровых изделий (на примере конкретного предприятия).
35. Проектирование и разработка базы данных кафедры информационных технологий ТАТУ.
36. Проектирование и разработка базы данных ТАТУ ТТJ-3.
37. Проектирование и разработка базы данных сети супермаркета Kozzinka.uz.
38. Проектирование и разработка базы данных сельского врачебного пункта.
39. Проектирование и разработка базы данных районного медицинского объединения.

40. Проектирование и разработка базы данных народной приемной Президента Республики Узбекистан.
41. Проектирование и разработка базы данных магазина, занимающегося продажей золотых изделий.
42. Проектирование и разработка базы данных автосалона (на примере конкретного автосалона).
43. Проектирование и разработка базы данных махаллинских сход граждан.
44. Проектирование и разработка базы данных предприятия по производству трикотажных изделий.
45. Проектирование и разработка базы данных салона красоты (на примере одного салона).
46. Проектирование и разработка базы данных кредитного отдела Silk Road Bank.
47. Проектирование и разработка базы данных ТАТУ.
48. Проектирование и разработка базы данных системы онлайн-регистрации для «O'zbekturizm».
49. Проектирование и разработка базы данных детской больницы для детей с ограниченными возможностями.
50. Проектирование и разработка базы данных отдела банка по работе с физическими лицами (на примере одного банка).
51. Проектирование и разработка базы данных отдела информационных технологий.
52. Проектирование и разработка базы данных деканата.
53. Проектирование и разработка базы данных предприятия по производству строительных материалов (на примере конкретного предприятия).
54. Проектирование и разработка базы данных компьютерной сервисной мастерской (на примере конкретного сервиса).
55. Проектирование и разработка базы данных районной аптеки.

56. Проектирование и разработка базы данных факультета вычислительной техники.
57. Проектирование и разработка базы данных информационной системы кассы Ташкентского метрополитена.
58. Проектирование и разработка базы данных системы онлайн-платежей ЖКУ.
59. Проектирование и разработка базы данных информационной системы железнодорожной кассы.
60. Дизайн и разработка базы данных Фонда Нурани.
61. Проектирование и разработка базы данных магазина канцтоваров (на примере конкретного магазина).
62. Проектирование и разработка базы данных системы прямой электронной торговли сельскохозяйственной продукцией.
63. Проектирование и разработка базы данных кафедры иностранных языков ТАТУ.
64. Проектирование и разработка базы данных информационной системы Джизакского аккумуляторного завода.
65. Проектирование и разработка базы данных компьютерного магазина.
66. Дизайн и разработка базы данных отдела кадров Капиталбанка.
67. Проектирование и разработка базы данных района.
68. Проектирование и разработка базы данных городских парков развлечений.
69. Проектирование и разработка базы данных нефтегазодобывающего предприятия.
70. Проектирование и разработка библиотечной базы данных.

ГЛАВА 2. ОРГАНИЗАЦИОННЫЕ АСПЕКТЫ ПРОЕКТНОЙ РАБОТЫ

2.1. Организационная работа при выполнении проектной работы

Темы для проектной работы задает руководитель, назначаемый кафедрой. Руководитель дает общие указания, направления по выполнению работы, дает советы по решению некоторых теоретических и практических задач, показывает необходимые руководства и литературу, которые можно использовать, контролирует выполнение проектной работы.

После того, как будет полностью завершено выполнение проекта, он будет передан руководителю на рецензию. После рассмотрения проектной работы руководителем и внесения им при необходимости исправлений, проектная работа выставляется на защиту. Защита проектной работы проходит перед преподавателями, назначенными кафедрой (при участии руководителя).

Во время защиты студенты готовят чертежи с пояснительной запиской, освещают актуальность выполненной работы, для какого типа предприятия она предназначена, принятые технические решения и предлагаемые технические и программные задачи с кратким изложением и с помощью ответов на вопросы.

Подводятся итоги по изложенным материалам и оценивается работа, выполненная студентом.

В случае если, проектная работа не будет подготовлена в срок, и она не будет защищена с положительной оценкой, она будет рассмотрена на заседании кафедры и будут приняты соответствующие меры в отношении студента.

2.2. Пояснительная записка проектной работы

Проектная работа состоит из пояснительной записки, чертежей и дополнительной информации. Пояснительная записка является основной частью проектной работы. Она включает введение, постановку задачи и

назначение, преимущества и недостатки методов решения рассматриваемой задачи, порядок решения задачи, анализ полученных результатов и заключительную часть.

Проектная работа состоит из следующих частей:

Титульный лист:

- На титульном листе пишутся название вышестоящей организации образовательного учреждения и название учебного заведения, а также название факультета и кафедры, на которой выполняется проектная работа, прописными буквами;
- На титульном листе оставляется отдельная строка для записи тему проектной работы, ФИО исполнителя, названия кафедры, защищающего проект, ФИО членов комиссии, показатель оцениваемого проекта.

Лист задания:

- На листе задания проектной работы название вышестоящей организации образовательного учреждения и название учебного заведения и название факультета и кафедры;
- После вышензложенного в правой части листа вставляется строка для подписи утверждения заведующей кафедры;
- Пишется название проекта, группа студента, выполняющего проект, имя и фамилия руководителя проекта;
- Представляется график выполнения проектной работы, т.е. указываются даты выполнения по главам проектной работы (дата начала и завершения);
- В самой нижней части листа задания ставится строка для даты задания и подписи руководителя проекта.

Содержание:

- Содержание проектной работы пишется на отдельной странице;
- Содержание должно включать введение, 3 главы, заключение и использованную литературу и приложения;

- В содержании должно быть указано номера страниц глав и параграфов каждой главы;
- Заключение и использованная литература также пишется на отдельной странице и номер страницы указывается в содержании;
- Нумерация приложений не обязательно, но в содержании должен быть указан номер главной страницы.

Вводная часть:

- Во вводной части проектной работы описывается область, где выполняется работа, актуальность проблемы, политическая и социально-экономическая значимость данных проблем, то есть будут изучены отражение их в соответствующих законах и постановлениях, и представлены во вводной части;
- Вводная часть должна состоять из 2-3 страниц и содержать точные данные;
- Содержание проектной работы должно быть отражено во вводной части, а также предоставлена соответствующая информация по выбранной теме;

Постановка задачи:

- В проектной работе постановка задачи имеет первостепенное значение, то есть должно быть обобщено сущность выполняемой работы.
- Задачи, подлежащие реализации в рамках проектной работы, приводятся последовательно и излагаются в главах;
- Задания, записанные в постановке задачи, должны соответствовать выполненным работам в главах;

Реализуемые работы в рассматриваемой области (краткий анализ):

- В рамках темы проектной работы осуществляется краткий анализ рассматриваемой области, изучаются существующие проблемы и их следует рассмотреть в рамках проектной работы;

- Анализ должен быть основан на реальных данных, и на использованные данные должны быть приведены соответствующие ссылки;

Заключение:

- Заключительная часть проектной работы должна быть написана на отдельной странице, а номер страницы должен быть указан в содержании;

- Заключительная часть должна быть написана по поставленным задачам на основе глав проектной работы, то есть окончательные выводы выражены на основе проделанной работы по главам;

- Результаты, описываемые в заключении, должны обосновать выполнение заданий, указанные в постановке задачи;

Использованная литература:

- Раздел проекта «Использованная литература» также должно начинаться с отдельной страницы и каждая литература должна быть пронумерована;

- В разделе «Использованная литература» в первую очередь излагаются соответствующие законы и решения в данной области, приказы вышестоящей организации;

- Должна быть правильно указана последовательность литературы по годам издания, то есть самая новая литература должна быть представлена в верхней части списка;

- Правила пользования каждой литературой должны быть изложены в правильном порядке: название литературы, автор, год и место издания и используемые номера страниц;

- В разделе «Использованная литература» также представлены случаи использования интернет-сайтов, в таких случаях полностью указывается адрес ссылки, по которой была получена информация с сайта;

Приложения:

- В разделе «Приложения» представляются таблицы, чертежи, результаты анализа (диаграммы и графики) и рисунки;
- Рабочие интерфейсы (GUI) и программные коды программного обеспечения, выполняемого в рамках проектной работы, также должны быть приведены в приложении;
- Нумерация приложений не обязательно, а дополнительная информация предоставляется исходя из общего объема проекта.

Цели и задачи проектной работы:

- теоретический, практический расчет, технический, алгоритмический, программный, опыт и другие необходимые аспекты решения основной задачи;
- представление необходимого технического обеспечения для решения задачи;
- Заключение и заключительная часть;
- Использованная литература;
- Дополнительная информация.

В разделе «Чертежи» чертежи, необходимые для пояснительной записки, выполняются в формате А4 (594x841 мм) в соответствии с существующими требованиями, их количество выбирается для полного раскрытия темы и в ряде случаев определяется руководителем.

Данная часть проектной работы может включать:

Структурная схема ИТ:

- структура информационного, математического, алгоритмического, программного, технического и иного обеспечения ИТ, в которых проводится анализ организационной, функциональной и иной структурной схемы;
- структурная схема информационных потоков;
- функциональная схема базы данных;
- блок-схема алгоритмов;

- структура, блок-схема программы, листинг полученных результатов, диаграммы и т.д.

В дополнительную часть включена дополнительная информация, не отраженная в приложениях, пояснительной записке или чертежной части. Сюда входят литературные сведения, методики расчета, инструкционные материалы и технологические схемы, чертежи, предназначенные для пояснения отдельных частей работы.

2.3. Структура проектной работы и ее основные части

После получения задания на проектную работу, студенты начинают работу с определения ее структуры и содержания, порядка выполнения, объема и времени для теоретической и практической части, расчетных, экспериментальных и чертежных работ и сбора первичных данных необходимых для её выполнения.

В ходе выполнения этого процесса студент упорядочивает всю необходимую информацию, собирает в основном информацию, приведенную в литературе и научно-технических статьях, отчет, написанный в результате научного исследования, и другую необходимую информацию.

Составляется список литературы и в него включаются необходимая учебная литература, учебные пособия, монографии, методическое пособие и др. для решения задачи, поставленной в проектной работе.

Помимо указаний кафедры и научного руководителя, студент должен будет обратиться к библиотекам, журналам и другим изданиям.

При сборе первичных данных, необходимых для проектной работы, студент должен:

1. Выполнить краткий обзор о выполненных работах по указанной теме в проектной работе;
2. Собрать литературу по выбранной теме;

3. Ознакомиться с необходимыми методами, алгоритмами и программами для её выполнения;

4. Уметь пользоваться техническими средствами и изучить методы вычисления с их помощью.

2.4. Порядок написания введения и основной части

Пояснительная записка является основным документом проектной работы и на его основании выходят на защиту. Вся работа, проделанная студентом по собранной или заданной теме, отражается в нем, освещается на достаточном уровне. В начальной части проектной работы пишется введение. В нем описывается развитие науки и техники, создание новых информационных и коммуникационных технологий, их значимость и необходимость на сегодняшний день. Кратко излагаются постановка задачи, выбранная гипотеза и задачи, необходимые для ее исследования, и будет целесообразно привести данные о необходимых методах расчета, моделях и алгоритмах, программ и других сведений для решения этих задач.

В основной части необходимо всесторонне освещать технический процесс по выбранной теме и автоматизируемое предприятие или систему или иной объект, его предыдущее и настоящее состояние, провести анализ его деятельности, а также необходимо подчеркнуть степень важности поднятого вопроса в выбранной теме.

В начале основной части даются отзывы о научных и практических исследованиях, связанные с выбранной темой, методах, алгоритмах и созданных программ, о возможностях их использования или усовершенствования, а также комментарии к литературе в этой области. В основной части описывается положительные и отрицательные стороны выбранной темы или результата, приводятся обоснования выбора метода, методики, закономерностей, алгоритмов и программ для решения поставленной задачи, необходимо показать преимущество выбранного метода перед другими.

В проектной работе студент должен описать теоретические аспекты выбранного метода, методологии, алгоритма или задачи, необходимых для проектирования и создания других функциональных или вспомогательных компонентов, необходимых для ИТ.

Затем, акцентируя внимание на ее практических аспектах, подчеркивается порядок решения информационных, математических, технических и других аспектов ИТ для конкретной задачи или объекта. При этом, будут приведены алгоритмы и программы вычислительных работ и проведен анализ полученных результатов с их помощью.

В проектной работе от студента требуется в процессе выполнения работы, связанная с алгоритмом и программой, обратить внимание на следующее:

- Математическая формулировка решаемой задачи;
- Создание алгоритма и составления блок-схемы программы для решения задачи;
- Выбор языка программирования, соответствующий алгоритму;
- Получение решения, выполнив скомпилированную программу и анализ полученного результата;
- Предоставление пользователю руководство по использованию программы;
- Сравнительный анализ предыдущего алгоритма и вновь созданных алгоритмов;
- Представление технологии проекта и др.

В заключительной части проектной работы приводится краткое описание выполненной работы по выбранной теме, выводы по полученным результатам, указывается уровень возможности работы по этим темам в будущем и чему следует придать значение при ее выполнении.

Кроме этого, приводится список используемой литературы.

Ниже приводится обобщенный вид сокращенных разделов для выполнения проектной работы.

Краткое описание процесса выполнения проектной работы:

1. Согласно выбранного варианта осуществляется сбор всей информации по отдельности, группируются по времени и анализируются.
2. Сгруппированные данные отображаются в виде графика и определяются точки пересечения групп.
3. Проводится анализ выбранной отрасли и определяются методы проектирования
4. При необходимости создается необходимая система на основе проектирования.
5. На основе созданного определения создаётся гипотеза и проводится ее исследование.
6. Результаты и выводы.

ГЛАВА 3. ПРАВИЛА ПОДГОТОВКИ ПРОЕКТНЫХ РАБОТ И ТРЕБОВАНИЯ К НИМ

Одной из основных задач в проектной работе является подготовка его в форме, отвечающей определенным требованиям. Материалы, относящиеся к выполнению проектной работы, даются в заданной последовательности.

Пояснительная записка пишется на узбекском, русском или иностранном языках на листе стандартного формата (210x297 мм) и должно соответствовать требованиям стилистики и грамматических правил.

При написании пояснительной записки необходимо обратить внимание на следующее:

- отступы: с левой стороны листа - 30 мм, с правой стороны – 10 мм, снизу и сверху - 20 мм;
- в верхней части письменного листа указывается последовательность страниц цифрами, отсчёт страниц начинается с обложки (на обложке не ставится) и проставляется до конца файла в едином порядке;
- таблицы и рисунки, включенные в пояснительную записку, включаются в этот счет, а номера обозначаются арабскими цифрами;
- объём пояснительной записки составляет 25-35 страниц, все формулы и страницы должны быть написаны четком виде;
- номер и название во всех приведенных рисунков указываются внизу страницы;
- порядковый номер и название приведенных таблиц указываются наверху страницы;
- порядковый номер математических выражений (на которые делается ссылка) указывается в скобках в правой части текста в виде (2), (II);
- работу необходимо подготовить и сшить в последовательности, указанной в пункте 2.2; в тексте могут использоваться общепринятые сокращения, в других случаях словосочетание приводится сначала

полностью, а сокращение приводится в скобках, например: Управление базой данных (УБД).

3.1. Требования к чертежам

Чертежи в проектной работе могут быть в виде рисунков, чертежей, диаграмм, графиков, таблиц и т.д. Это должны быть чертежи, помогающие наглядно и подробно показать пункты, написанные или зафиксированные в пояснительной записке. Каждый чертеж должен иметь подпись с указанием его названия, и эта подпись должна быть написана внизу чертежа. На чертежах проставляются номера и эти номера отмечаются в тексте.

3.2. Требования к вводу документов, связанных с программами

При решении задачи, связанной с той или иной частью информационных систем, необходимая вычислительная работа часто связана с программой. Данные программные работы должны быть представлены в качестве руководства или инструкций по использованию для других. Это особенно важно для специальностей по направлению «ИТ-сервис».

Должен соответствовать требованиям для программ, программных пакетов или приложений.

Если проектная работа связана с программами, в ее основной части должны быть приведены следующие данные:

- общие (название, на каком языке программирования);
- для решения каких задач предназначена программа;
- порядок использования;
- использованное техническое обеспечение;
- входные данные;
- выходные данные;
- сравнительный анализ созданной программы с аналогичными программами.

3.3. Порядок составления списка используемой литературы

Список литературы, использованной при выполнении проектной работы, приводится в конце пояснительной записки. Список отображается в определенном порядке, то есть последовательном порядке. При составлении этого списка рекомендуется придерживаться следующих правил.

В каждой приведенной литературе указываются фамилия автора, инициалы имени и отчества, название книги или статьи, город и издательство издания, год, номер сборника, страницы. В списке указывается только использованная литература, и они должны быть пронумерованы в пояснительной записке цифрами. Цифровые номера, обозначающие литературу в пояснительной записке, заключаются в квадратные скобки.

ГЛАВА 4. ПОРЯДОК ЗАЩИТЫ ПРОЕКТНОЙ РАБОТЫ

4.1. Подготовка к защите проектной работы

После того, как проект по заданной теме будет полностью выполнен, он будет защищен перед комиссией, созданной распоряжением заведующего кафедрой. Поэтому требуется подготовиться на соответствующем уровне. Для того чтобы студент мог объяснить истинный смысл проделанной им работы в процессе подготовки к защите, целесообразно отметить постановку задачи, методы её решения и проведенные им исследования по полученным результатам. Для этого он должен просто и понятно изложить выполненную им работу.

После доклада комиссия, принимающая проектную работу, и руководитель проверяют уровень знаний студента в этой отрасли, задав несколько вопросов, и выставляется итоговая общая оценка.

Данные работы учат студента самостоятельно решать поставленную задачу по выбранной специальности и чётко и понятно объяснять выполненную работу другим.

4.2. Порядок работы комиссии, принимающей проектные работы

Комиссия, принимающая проектные работы назначается заведующим кафедрой. Эта комиссия должна приходить в определенные дни недели до начала экзамена (сессии), а затем проводить защиту в назначенное время.

В состав этой комиссии входят руководители проектной работой и ведущие преподаватели кафедры. Присутствие заведующего кафедрой и представителя деканата не обязательно.

4.3. Защита проектной работы

Комиссия, утвержденная руководителем и кафедрой, собирается и принимает проектную работу. Для этого студент представляет копию

пояснительной записки с подписью научного руководителя и своей подписью.

После краткого доклада студента, переходят на вопросы и ответы. После этого руководитель характеризует проектную работу и способности студента в выбранной отрасли и по согласованию с членами комиссии выставляется общая оценка. Оценка выставляется по существующей рейтинговой системе. Если проектная работа признается неудовлетворительной, то есть исходя из доклада и ответов на заданные вопросы уровень знаний студента по заданной теме находят низким, проектная работа оценивается «неудовлетворительной». Соответствующий деканат будет уведомлен, и в отношении студента будут приняты меры.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. O'zbekiston Respublikasi prezidentining 2017 yil 7 fevraldagi PF-4947-son "O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida"gi Farmoni.
1. Mirziyoev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Toshkent. «O'zbekiston», NMIU, 2017. – 488 b.
2. В.П. Базы данных. Книга 2 Распределенные и удаленные базы данных: учебник.// Москва ИД «ФОРУМ» - ИНФРА-М. – 2018. – С 261.
3. Rahul Batra. SQL Primer An Accelerated introduction to SQL Basics.// Gurgaon, India. 2019. – P 194.
5. Мартишин С.А. Базы данных. Практическое применение СУБД SQL – и NoSQL – типа для проектирования информационных систем: учеб. Пособие // - Москва: ИД «ФОРУМ» - ИНФРА-М, 2019, – 368 с.
4. Aripov M., Begalov B., Begunkulov U. va boshqalar."Axborot texnologiyalari" / T.: "Noshir", 2009.-368 b.
5. Бегалов Б.А., Султанов С.М. "Информационные системы бухгалтерского учета": Проблемы, поиски, решения. Монография. – Т.: «Фан», 2009., 116 с.
6. Мусаев А.А., Бегалов Б.А. «Информационно-коммуникационные технологии в национальной экономике» Монография.- Т.: Фан, 2008.- 146 с.
7. Ayupov L.F., Begalov B.A. va boshqalar. "Shaxsiy kompyuterlar va ulardan samarali foydalanish asoslari". O'quv qo'llanma. TDIU.2007.
8. Informatika va axborot texnologiyalari. Akademik S.S. G'ulomov umumiy tahriri ostida. Darslik. "Iqtisodiyot" Toshkent. – 2009.
9. Informatika. Akademik S.S. G'ulomov umumiy tahriri ostida. Toshkent. – TDIU, 2007
10. Alimov R, Xodiev B, Alimov Q va boshqalar. "Milliy iqtisodda axborot tizimlari va texnologiyalari", T.: "Sharq"-2004
11. G'ulomov S.S, Alimov R.X, va boshqalar. "Axborot tizimlari va

texnologiyalari” “Sharq”, T.:-20 10.

12. Murodova., Tulyaev D.R. Ma'lumotlar bazalari va banklari. O'quv qo'llanma. TDIU.2010.
13. Alimov R.X., Tulyaev D.R. Intellektual tizimlar. O'quv qo'llanma. TDIU. 2010.

Приложение 1. Пример оформления титульного листа отчета о выполнении проектной работы для студентов по направлению «Компьютерный инжиниринг» («ИТ-сервис»)

**МИНИСТЕРСТВО РАЗВИТИЯ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Кафедра «Информационные технологии»

Проектная работа
по предмету «Инжиниринг услуг информационных технологий и управление
услугами».

по теме “-----
-----”

Выполнил: _____
(ФИО, группа)

Руководитель: _____
(ФИО)

Защита: _____
(дата)

(оценка)

(подпись)

(подпись)

Ташкент - 2022

Приложение 2. Пример написания отчетного листа-задания по выполнению проектной работы по предмету «Система управления базами данных» для студентов направления «Компьютерный инжиниринг» («ИТ-сервис»)

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛЬ-ХОРЕЗМИЙ

Факультет: «Компьютерный инжиниринг»

Кафедра: «Информационные технологии»

Направление: «ИТ-сервис»

Утверждаю

Заведующий кафедрой _____

« ____ » _____ 20__ г.

Задание на проектную работу

1. По проектной работе _____
2. Группа _____
3. ФИО студента _____
4. ФИО руководителя _____
5. Тема проектной работы _____
6. Тема утверждена на заседании кафедры. Протокол № “__” “__” 20__ год.
7. Первоначальная информация о работе: _____

№	Название	Начало	Завершение

8. Содержание расчетно-пояснительной записки (перечень вопросов, подлежащих проработке):

9. Перечень графических материалов:

Дата назначения: _____

Руководитель: _____

Приложение 3. Отчет о выполнении проектной работы по предмету «Система управления базами данных» для студента направления «Компьютерный инжиниринг» (ИТ-сервис)

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛЬ-ХОРЕЗМИЙ**

Факультет: «Компьютерный инжиниринг»

Кафедра: «Информационные технологии»

По предмету «Система управления базами данных»

Проектная работа

по теме:

«Проектирование и разработка базы данных больницы»

Выполнил: _____
(ФИО, группа)

Руководитель: _____
(ФИО)

Защита: _____
(дата)

(оценка)

(подпись)

(подпись)

Ташкент - 2022

**МИНИСТЕРСТВО РАЗВИТИЯ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**
**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Факультет: «Компьютерный инжиниринг»

Кафедра «Информационные технологии»

Направление (специальность): ИТ-сервис

Утверждаю

Заведующий кафедрой _____

« ____ » _____ 20__ г.

ЗАДАНИЕ

на проектную работу

По проектной работе:

Руководитель: _____

Тема проектной работы: «Проектирование и разработка базы данных больницы»

Тема утверждена на заседании кафедры протоколом №1.

Первоначальная информация о работе.

График выполнения работы:

№	Название	Начало	Завершение
1.	Анализ предметной области системы проектирования и разработки баз данных больницы	10.09.2020	10.10.2020
2.	Проектирование и организация базы данных больничного центра	10.10.2020	10.11.2020
3.	Инструкция по использованию разработанного программного обеспечения.	10.11.2020	10.12.2020
4.	Заключение	12.12.2020	25.12.2020

Содержание расчетно-пояснительной записки (перечень вопросов, подлежащих проработке): _____

Перечень графических материалов: _____

Дата назначения: _____ Руководитель: _____

СОДЕРЖАНИЕ

ВВЕДЕНИЕ		3
ГЛАВА 1.	ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ БОЛЬНИЦ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ РАЗРАБОТАННОЙ СИСТЕМЫ.....	8
1.1.	Анализ базы данных больницы.....	8
1.2.	Общие понятия о языке скриптов JavaScript и Node JS.....	11
1.3.	Управление базами данных NoSQL (нереляционными) и MongoDB.....	14
1.4.	Постановка задачи.....	16
	Выводы по главе 1.....	18
ГЛАВА 2.	ОРГАНИЗАЦИЯ СОЗДАНИЯ БАЗ ДАННЫХ БОЛЬНИЦЫ.....	19
2.1.	Организация информационной модели базы данных «Больница».....	19
2.2.	Создание базы данных больницы в среде MongoDB.....	21
2.3.	Подключение базы данных MongoDB к Node.js.....	29
	Выводы по главе 2.....	32
ГЛАВА 3.	ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	33
3.1.	Системы базы данных больницы и таблица данных.....	33
3.2.	Интерфейс разработанного программного обеспечения.....	36
	Выводы по главе 3.....	39
ОБЩИЕ ВЫВОДЫ.....		40
ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА.....		41
ПРИЛОЖЕНИЯ (код программы и программа)		42

ВВЕДЕНИЕ

Здравоохранение в Республике Узбекистан - это система социально-экономической и медицинской деятельности, направленной на охрану здоровья населения, здравоохранение включает в себя общие мероприятия, направленные на профилактику и лечение заболеваний, создание здоровых условий жизни и труда, обеспечение высокой трудоспособности и долгую жизнь; его основная задача - оказывать помощь пациентам современным, специализированным и надлежащим образом.

Здравоохранение имеет долгую историю на Востоке, особенно в Узбекистане. Особенно в средние века в этом регионе был достигнут высокий уровень в сфере медицины. В этот период были переведены на арабский язык древние медицинские труды с греческого и других языков, а содержащиеся в них рекомендации применялись на практике. В то же время мыслители Востока создали общемировые труды по медицинской науке. В этом аспекте известны научные труды как, «Фармакология в медицине» («Kitob assaydana fittib») Абу Райхана Беруни и «Законы медицины» («Al Qonun fittib») Абу Али Ибн Сины, которые долгое время были основным руководством в мировой медицине. Особенно Абу Али Ибн Сина прославился в свое время как врач (хаким), использующий новые методы лечения. В более поздние времена основным видом здравоохранения в Туркестане стала лекарство. Лекари учились в медресе или путем наставник-ученик учились лечить различные болезни, переломы, раны, предотвращать болезни. Многие из них занимались не только лечением больных, но и приготовлением лекарств. Амир Темур, среди прочего просветительской деятельности, уделял особое внимание здравоохранению населения. Пригласил известных лекарей из других стран в столицу Самарканд и открыл специализированные больницы.

В 8-9 вв. в Мовароуннахре и Хорасане начали строить специальные здания для госпиталей. В 15 веке в таких городах, как Самарканд и Бухара,

было построено много больниц. Такие государственные больницы назывались «Dor ushshifo» («Дом врачевания»), где лечили больных знатные лекари того времени. Фармацевты готовили лекарства для больных в специальных аптеках «Dor ushshifo». В большинстве больниц, кроме лечения больных, лекари занимались и медицинской наукой. Алишер Навои также придавал большое значение открытию больниц. В годы войны 1941-45 гг. перед системой здравоохранения Узбекистана стояла очень ответственная задача. Необходимо было открыть госпитали для раненых и пострадавших бойцов, еще больше укрепить материально-техническую базу системы здравоохранения, поддерживать стабильное санитарное состояние страны. Вскоре было открыто более 130 военных госпиталей, Военно-медицинская академия в Ленинграде, Московский медицинский институт и другие медицинские учреждения были переведены в Узбекистан.

В республике были открыты специальные хирургические госпитали для лечения инвалидов войны. После войны в республике был перезапущен ряд медицинских НИИ и появились новые (санитарии, гигиены и профзаболеваний, онкологии и радиологии, педиатрические НИИ и др.). Были организованы в 1955 году Андижанский медицинский институт, в 1972 году Среднеазиатский, ныне Ташкентский медицинский педиатрический институт, а в 1990 году был открыт Бухарский медицинский институт. Позже были сформированы Ургенчский филиал 1-го ТашТИ, Ферганский филиал 2-го ТашМИ и Нукусский филиал Ташкентского медицинского педиатрического института. Открыты крупные клиники, оказывающие услуги в области кардиологии, гастроэнтерологии, эндокринологии, неврологии, гематологии, пульмонологии, аллергологии, ревматологии и других областях. Количество больниц в Узбекистане составляло 1010, а количество коек в них – 140072. При советской власти сеть медицинских учреждений была значительно расширена, но из-за экологического дисбаланса наблюдался рост различных внутренних болезней среди

населения в результате разрушения внешней среды, в частности влияния различных сильных химикатов на землю, монополию хлопка и прочие социальные неприятности.

После обретения Узбекистаном независимости вновь началось реформирование сферы здравоохранения, основным принципом системы здравоохранения является восстановление профилактического направления; разработка и внедрение ее современных и эффективных методов; усиление мер по здравоохранению матери и ребенка; совершенствование системы медицинского образования и подготовки кадров; реорганизация работ информационных технологий исходя из требований эпохи и уровня жизни общества; создание в стране отрасли, производящую качественные и в достаточном количестве фармацевтические препараты, медицинские товары и технические средства, относящиеся к этой сфере.

Была поставлена основная задача, как дальнейшее совершенствование материально-технической базы учреждений здравоохранения; выведение часть лечебно-профилактических учреждений из государственной собственности и изменение формы собственности.

В Узбекистане работы по здравоохранению осуществляются под управлением Министерства Здравоохранения Республики Узбекистан. В Республике Каракалпакстан имеются соответствующие управления или отделения в Министерстве здравоохранения, областях, городах и районах. Федерация профсоюзов Республики Узбекистан, а также Международный благотворительный фонд «За здоровое поколение», Общество «Красный полумесяц», «Нурони» вносят свой вклад в развитие системы здравоохранения Республики.

Переход к рыночным отношениям также предъявил новые требования к сфере здравоохранения. В частности, снижение затрат на административное управление; использование экономичных и эффективных методов при лечении больных; перевод части объема работы больниц в поликлиники,

сокращение количества малоэффективных вакансий; осуществляется оплата работ, выполненных на основании договора, и другие меры....

Актуальность темы. Целью реформ, проводимых в нашей республике за годы независимости, является вхождение в ряды развитых стран и повышение уровня жизни народа на более высокий уровень. Достижение этой цели невозможно представить без развития системы здравоохранения. Но для достижения этой цели важное место занимают большие средства и государственная политика в этой сфере. Несомненно, нам важно изучать нашу прошлую и ближайшую историю и делать из нее выводы. Потому что, как сказал наш Президент, «Если каждая нация, каждый народ сохранит свою родословную, будущее поколение и будет знать себе цену, то великое будущее того народа и того государства неизбежно.

Известно, что наша страна сначала страдала от произвола царского колониализма, а затем от гнета советского авторитарного режима. В этой эпохе благодаря политике колониализма и насилию над нашим народом уровень здоровья нации была доведена до края обрыва. В частности, характерной чертой этих строев является стремление к богатству, действия для его приобретения вопреки любым потерям. Неправильная политика в отношении Аральского моря, веками сохранявшего стабильным и умеренным природный климат нашей страны, в конечном итоге привела к трагическому завершению судьбы столь уникального дара природы. В бассейнах двух больших рек Амударьи и Сырдарьи масштабные работы в области нового освоения земель потребовали дополнительного забора речной воды. В результате, наряду с выращиванием хлопка, в некоторых регионах Узбекистана получило развитие рисоводство.

Однако из-за того, что освоение земель не осуществлялось на научной основе, а развитие сельского хозяйства велось экстенсивным путем, это привело к чрезмерной и неоправданной трате водных ресурсов. Вода не доходила до острова в необходимом количестве, из-за чего море высыхало, а

большое количество соли из засушливых районов разносилось ветром и штормами. В результате усиливался процесс непригодности земель в регионах к земледелию, готовились условия для появления факторов, облегчающих заражение населения различными заболеваниями. Эпидемиологические и санитарно-гигиенические условия на острове также ухудшились, что привело к росту заболеваемости и смертности людей.

К 1991 году количество больниц в нашей стране составляло 1370. Хотя система здравоохранения при советской власти расширилась, среди населения увеличились различные заболевания в результате внедрения различных сильнодействующих химикатов, монополии на хлопок и других социальных последствий. В нашей стране ежегодно миллионы людей обращаются в медицинские центры по поводу болезни, и определенное количество из них заражаются инфекционными заболеваниями. Хуже всего то, что большинство больных инфекционными заболеваниями были младенцами.

В это время уровень смертности среди матерей также поднялся до гораздо более высокого уровня. Средняя продолжительность жизни людей сильно отставала от развитых стран мира. Политика, проводившаяся в период советского самодержавия, привела к тому, что отчетливо стал виден не только застой экономической жизни, но и упадок в социальной и духовной сфере. Финансирование системы здравоохранения стало второстепенным вопросом в условиях экономического застоя, отставания экономики Союза от уровня развитых стран и на этой почве увеличился дефицит государственного бюджета. В частности, мы можем наблюдать это на примере Узбекистана, его отдаленной колониальной страны.

Цель проектной работы. Основным вопросом при подготовке данной проектной работы является создание базы данных больницы. То есть облегчить работу сотрудников больницы и создать электронную информационную систему больницы, не создавая нарушений в их системе.

Формирование навыков использования MongoDB в качестве базы данных, а также программных инструментов HTML, JavaScript, eJS, Node JS в ходе выполнения проектной работы.

ГЛАВА 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ БОЛЬНИЦ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ РАЗРАБОТАННОЙ СИСТЕМЫ

1.1. Анализ базы данных больницы

В качестве предметной области данной проектной работы была выбрана база данных больницы. При разработке базы данных больницы необходимо изучить процедуры принятия в стационар новых и опытных врачей-специалистов, увольнения действующих, регистрации новых пациентов и хранения данных вылеченных пациентов в архиве, фиксирования закрепленных за ними врачей, назначенных препаратов, и ведения учета лекарств, доставленных в стационар, стиль, структуру и работу онлайн-систем для системы управления. На основе изученной информации будет спроектирована База данных.

Задача предметной области проектной работы заключается в формировании базы данных для ведения базы данных больницы, хранения базы данных, при необходимости, измененных и удаленных данных, и автоматизация работы системы, для удовлетворения потребностей пациентов в короткие сроки, облегчения труда сотрудников, ускорения процессов взаиморасчетов и устранения излишней бумажной волокиты и т. д.

1. При работе с конкретной информацией человек ограничивается той или иной областью реального мира. В таких случаях только некоторые объекты вызывают интерес для изучения. Совокупность таких объектов называется предметной областью.

1. Никто не может отрицать тот факт, что мы сейчас живем в век информационных технологий. Однако, мы храним большую часть нашей личных данных или данные, связанные рабочей деятельностью, в электронном виде. Кроме того, большая часть населения планеты использует в повседневной жизни Интернет, содержащий огромное количество информации. По мере увеличения объема информации, которую мы используем, управлять ею становится все сложнее. Естественно возникает

вопрос, как поисковые сайты в Интернете могут найти информацию, с её тысячами альтернативами, которую мы ищем, за секунды? Каким образом электронные информационные системы, состоящие из набора нормативно-правовых документов, могут мгновенно находить нужные документы? Ответ прост, все это осуществляется через системы управления базами данных (далее СУБД). Для СУБД разработано множество программных решений, например, реляционных: SQL Server, Oracle, MySQL, MS Access и нереляционных: MongoDB, MariaDB и др.

При разработке базы данных использовались 4 таблицы:

- Сотрудники
- Пациенты
- Лекарства
- Поставщики.

Эти таблицы имеют столбцы, как показано в следующей таблице:

Табл. 1.1. Предметная область базы данных больницы.

Сотрудник	<ul style="list-style-type: none"> - ФИО - Телефон - Должность - Фотография - Зарплата - Адрес - Пациенты - Рабочее время
Пациент	<ul style="list-style-type: none"> - ФИО - Телефон - Фотография - Возраст - Доктор - Лекарства

	<ul style="list-style-type: none"> - Адрес - Время прибытия
Лекарства	<ul style="list-style-type: none"> - Название - Поставщик - Тариф (тип) - Стоимость - Количество - Фотография
Поставщик	<ul style="list-style-type: none"> - Название - Фотография (logo) - Тип лекарства

В результате глубокого анализа системы больницы я решил сформировать предметную область базы данных больницы, как указано выше и в ходе создания онлайн-системы для больницы использовал одну из современных нереляционных базы данных - MongoDB, и при создании программного обеспечения - языка HTML, EJS и JAVASCRIPT. Связь между базой данных и программным приложением осуществляется с помощью Node JS.

1.2.Общее понятие о языке сценариев JavaScript и Node JS

JavaScript - это язык программирования, основанная на текст, используемый как клиентом, так и сервером для создания интерактивных веб-страниц. В то время как HTML и CSS - это языки, придающие структуру и стиль веб-страницам, JavaScript предоставляет веб-страницам интерактивные элементы, привлекающие пользователя. Простые примеры JavaScript, которые вы можете использовать каждый день, включают платформу для поиска на Amazon, видеобзор новостей, размещенный в The New York Times, или обновления в Twitter.

Добавление JavaScript улучшает представление пользователя, превращая веб-страницу из статической страницы в интерактивную. JavaScript придает веб-страницам динамичный характер, добавляя передвижение и анимацию. JavaScript в основном используется для веб-приложений и веб-браузеров. Но помимо Интернета JavaScript также используется для управления программным обеспечением, серверами и встроенным оборудованием.

Разработчики могут использовать различные фреймворки (framework) JavaScript для разработки и создания веб-приложений и мобильных приложений. Фреймворки JavaScript - это множество библиотек кодов JavaScript, которые предоставляют разработчикам предварительно написанный код для типовых функций и задач программирования, что позволяет им легко и быстро создавать веб-сайты или веб-приложения.

Популярные устаревшие фреймворки JavaScript включают React, React Native, Angular, Vue и другие. Многие компании используют Node.js - среду JavaScript, созданную с использованием языка программирования C++ на базе JavaScript V8 engine Google Chrome. Paypal, LinkedIn, Netflix и Uber — вот лишь несколько популярных примеров!

Помимо веб-сайтов и приложений, разработчики также могут использовать JavaScript для создания простых веб-серверов и разработки инфраструктуры с использованием Node js.

Node js - это серверная среда JavaScript с открытым исходным кодом, которая работает на всех платформах и выполняет код JavaScript вне веб-браузера. Node js позволяет разработчикам использовать JavaScript для написания инструментов командной строки и создания сценариев на стороне сервера — сценариев, запускаемых сервером для создания динамического содержимого веб-страницы до того, как страница будет отправлена в веб-браузер пользователя. Следовательно, Node.js представляет собой парадигму «JavaScript везде», которая объединяет веб-приложения вокруг одного языка

программирования, а не разных языков для сценариев на стороне сервера и на стороне клиента.

Хотя `.js` является стандартным расширением имени файла для кода JavaScript, имя «Node.js» в этом контексте не относится к конкретному файлу, а является просто названием продукта. Node.js имеет управляемую событиями архитектуру, поддерживающую асинхронный ввод-вывод. Эти варианты дизайна сосредоточены на оптимизации пропускной способности и масштабируемости для веб-приложений с большим количеством операций ввода-вывода, а также для веб-приложений, работающих в режиме реального времени (таких как приложения для связи в реальном времени и браузерные игры).

1.3. Об управлении базами данных NoSQL (нереляционными) и MongoDB.

Люди при использовании термина «база данных NoSQL», обычно под ним понимают использования любой не связанной (не реляционной) базы данных. Для некоторых термин «NoSQL» означает «не-SQL», для других — «не только SQL». Однако большинство людей согласны с тем, что базы данных NoSQL - это базы данных, которые хранят данные в формате, отличном от реляционных таблиц. Распространенным заблуждением является то, что базы данных NoSQL или нереляционные базы данных плохо хранят реляционные данные. Базы данных NoSQL могут хранить реляционные данные — просто они хранятся иначе, чем базы данных. На самом деле, по сравнению с базами данных SQL, многие люди считают, что моделирование реляционных данных в базах данных NoSQL проще, чем в базах данных SQL, потому что нет необходимости разделять соответствующие данные между таблицами.

Модели данных NoSQL позволяют размещать соответствующую информацию в структуре данных.

Базы данных NoSQL появились в конце 2000-х, когда резко упали затраты на хранение. Прошли те времена, когда была необходимость в создании сложной, трудно управляемой модели данных для уменьшения дублирования данных. Разработчики (а не хранилище) становятся основной статьей расходов при разработке программного обеспечения, поэтому разработчики оптимизируют базы данных NoSQL для повышения производительности.

На сегодняшний день самой популярной базой данных NoSQL является MongoDB. MongoDB - это кроссплатформенная программа для баз данных, основанная на документах. Классифицируемая как база данных NoSQL MongoDB использует документы, подобные JSON, с произвольными схемами. MongoDB разработана MongoDB Inc. и лицензирована под Server Side Public License (SSPL).

MongoDB отлично подходит для транзакционных хранилищ, что является хорошим выбором, когда вам нужна скорость или в будущем будет необходимость внести много изменений. MongoDB отлично подходит для поддержания развития содержимого данных с течением времени, поскольку его бессхемные операции позволяют быстро обновлять данные.

MongoDB легко установить и настроить. Основная особенность MongoDB заключается в том, что это бессхемная база данных. Нет больше миграции схемы. Поскольку MongoDB не имеет схемы, ваш код определяет вашу схему. Возможность создания модели данных на основе документов — одно из самых привлекательных преимуществ MongoDB. Метод хранения данных в виде BSON (Binary JSON), рубиновых хэшей и т.д. помогает хранить данные очень богатым виде, с возможностью хранения массивов и других документов. Язык запросов документов, поддерживаемый MongoDB, играет важную роль в поддержке динамических запросов. Из-за способа структуризации данных в MongoDB комплексные джоины (join) не требуются (формат BSON — пара ключ-значение).

1.4. Постановка задачи

Основными задачами системы базы данных больницы являются повышение уровня эффективности работы персонала больницы, снижение затрат, облегчение снабжения лекарствами, снижение счетов, своевременное и качественное обслуживание пациентов. Для этого требуется качественная и хорошо спроектированная база данных. Создаваемая база должна обеспечивать всесторонний комфорт пользователям. Данная база данных больницы в несколько раз облегчит работу сотрудников.

База данных и программное обеспечение больницы, созданных по инициативе администрации больницы, могут повысить точность работы персонала и избежать трудоемкой бумажной работы, улучшить и оценить работу персонала, контролировать, насколько хорошо пациенты получают лечение и их текущее состояние, а также определять количество лекарств на складе и помогают оперативно заключать договоры на основе удобных и точных расчетов с компаниями и фирмами, поставляющими лекарства, и не допускать дефицита лекарств и необходимого оборудования.

В целях повышения эффективности работы и снижения излишних расходов целесообразно внедрить такие онлайн-системы баз данных не только во всех больницах, но и во всех государственных и негосударственных организациях, производственной сфере и государственных органах нашей страны.

Если система организована неправильно, то естественно, что в системе будут ошибки и недоразумения. Мы должны предотвращать появление дефектов и ошибок и шаг за шагом изучать систему.

В норме проект может состоять из одной схемы, но на практике использовать его сложно, так как, как правило, коммуникативные процессы очень сложны и занимают много времени. Такая модель процессов включает:

- Объекты и процессы. Объекты представляют ресурсы, используемые в процессах.

- Цели моделирования информационных систем формулируются из следующих соображений:

- понимание состава системной базы данных больницы и обеспечивать динамику происходящих в ней процессов;

- понимание текущих проблем системы баз данных больниц и обеспечение возможностей для их решения;

- создание базы для формирования требований к программному обеспечению, автоматизирующему проектирование системы базы данных больницы.

Все программные средства, которые будут установлены в ближайшие годы и будут созданы в будущем, должны полностью отвечать требованиям этой системы.

Цель проектной работы – создать необходимую для населения нашей страны автоматизированную систему, используя вышеперечисленные требования и возможности, созданные для персонала, а также создать для них удобства на достаточном уровне.

Можно с уверенностью сказать, что тот факт, что эта система построена на MongoDB, которая в настоящее время имеет высокий уровень безопасности и эффективности, доказывает, насколько важна эта система.

Выводы по главе 1

В вводной части своей проектной работы я постарался показать актуальность темы. В основном была предоставлена информация о системе баз данных больницы. Были рассмотрены вопросы, связанные с анализом предметной области, постановкой задачи и проектом. Рассмотрены этапы проектирования, общие понятия о языке программирования JavaScript и Node JS, алгоритм решения задачи, общие концепции баз данных NoSQL и

MongoDB. Дано подробное описание об информации, включенная в анализ предметной области.



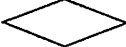
В постановке задачи определены следующие вопросы, то есть какую информацию о выбранных объектах необходимо включить, какие области необходимо осветить. Представлена последовательность работ на этапах проектирования. В процессе подготовки данной проектной работы у меня появилось множество представлений по работе с реальными проектами и на что следует обратить внимание в них.

ГЛАВА 2. ОРГАНИЗАЦИЯ СОЗДАНИЯ БАЗЫ ДАННЫХ БОЛЬНИЦЫ

2.1. Организация инфологической модели базы данных больницы.

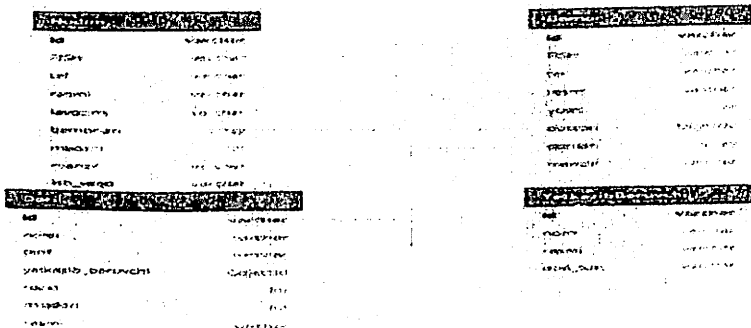
В первую очередь нам важна роль блок-схем и связей при создании инфологической модели. Функции блок-схем приведены ниже.

Таблица 2.1 Элементы в структуре базы данных «Больница».

	С помощью этого блока определяется форма диаграммы.
	С помощью этого блока определяются Атрибуты, и они соединяются с объектами с помощью не направляющих линий.
	При помощи этого блока определяется связи между объектами.
<----->	Этот символ используется для обозначения связей 1-к-1 (1:1).
<----->>	Используется для представления связей «многое к 1» (M:1).
<<----->>	(M:N) представляет связь «многое ко многим».

Построим инфологическую модель, используя приведенные выше блоков.

Рисунок 1.1. Модель структуры базы данных больницы.



2.2. Создание системы базы данных больницы в среде MongoDB.

Я использовал систему управления базой данных MongoDB для создания системы базы данных больницы. В системе управления базами данных мы используем специальные команды MongoDB и функцию сайта cloud.mongodb.com для создания, редактирования и удаления объектов. Для того, чтобы разработать систему базу данных больницы, мы в первую очередь создаем ее базу. Мы выполняем часть кодов в MongoDB и некоторые функциональности, используя следующее окно:

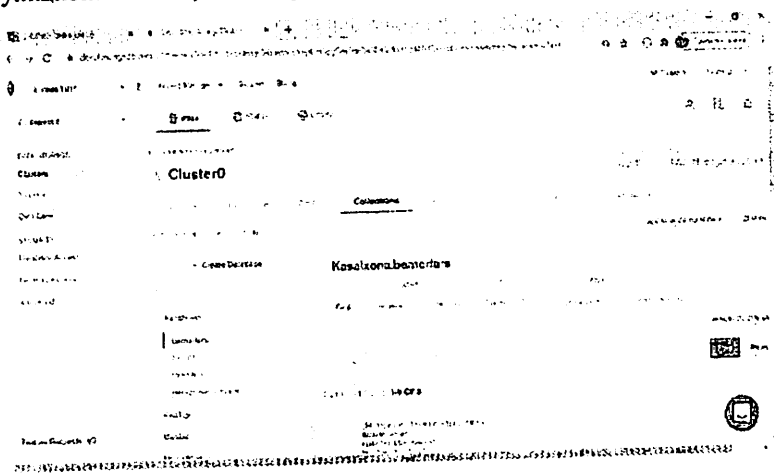


Рисунок 2.1. Окно системы базы данных больницы в MongoDB.

Создаем базу данных через окно персонального кабинета системы управления базами данных MongoDB на рисунке выше. Такие функции, как удаление, модификация и т.д. сформированной базы данных, мы выполняем, используя возможности персонального кабинета. Сначала мы создадим систему базу данных больницы (В данном случае мы уже создали эту базу данных. Новая база данных будет создана с помощью «+ Create Database»):

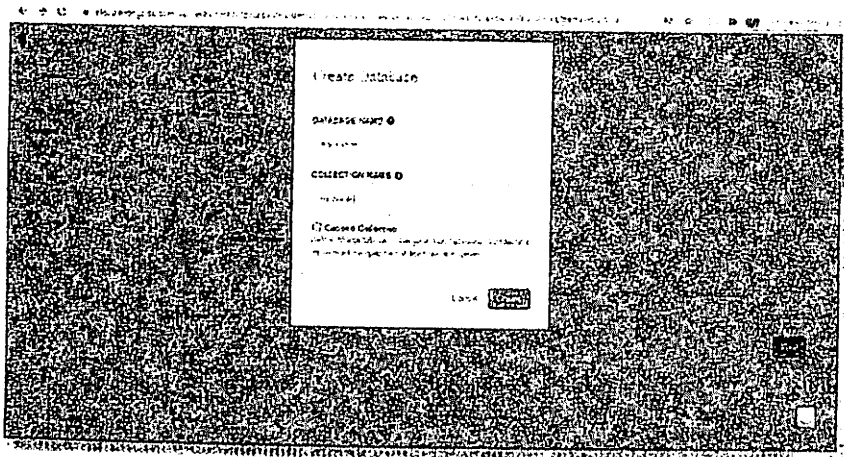


Рисунок 2.2. Создание системы базы данных больницы

После создания системы базы данных больницы мы создадим множество под названием «Сотрудники».

В этом множестве мы создаем разделы по сотруднику, такие как «id», «ФИО», «должность», «телефон», «фотография», «адрес», «зарплата», «пациенты» и «рабочее время». Для этого мы используем Node.js и создаем следующую модель:

```

const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const EmployeeSchema = new Schema({
  name: String,
  surname: String,
  position: String,
  phone: String,
  address: String,
  salary: String,
  patients: String,
  workTime: String,
  photo: String,
  id: String
});

const Employee = mongoose.model('Employee', EmployeeSchema);

module.exports = Employee;
  
```

Рисунок 2.3. Модель сотрудника в базе данных больницы

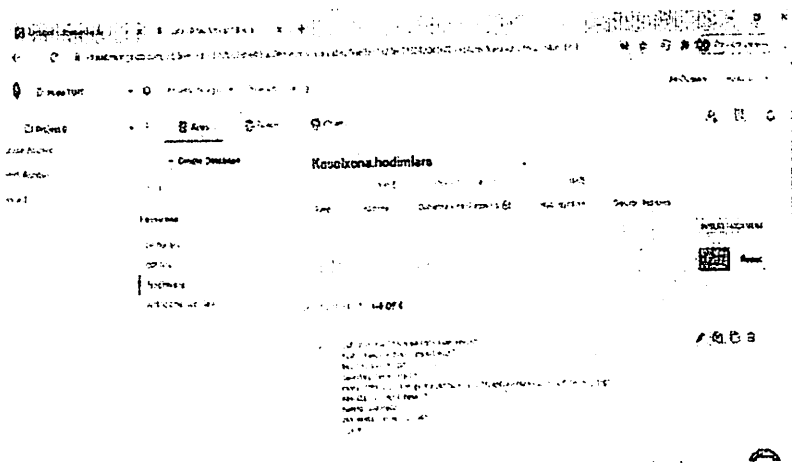


Рисунок 2.4. Множество сотрудников в базе данных больницы

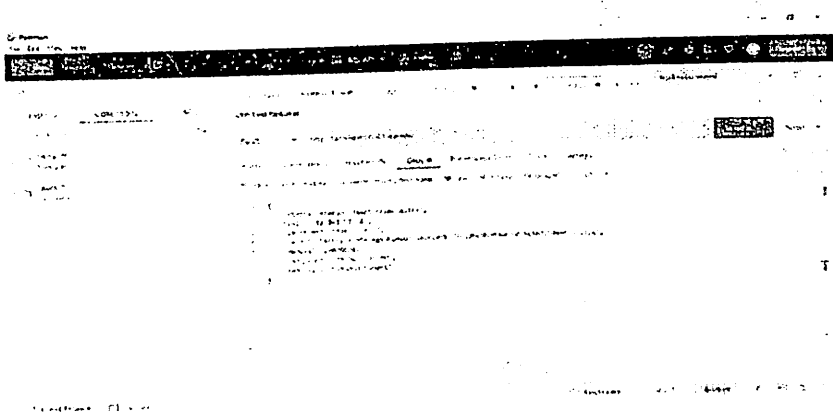


Рисунок 2.5. Добавление данных в множество сотрудников

После этого создаем множество «Поставщики» и создаем разделы «название», «фотография» (т.е. логотип) и «тип препарата»:

```

const mongoose = require('mongoose');
const { Schema } = mongoose;

const yetkazibberuvchilarni = new Schema({
  nomi: {
    type: String,
  },
  pasagi: {
    type: String;
  },
  dori_turi: {
    type: String,
  },
});

module.exports = mongoose.model('Yekazibberuvchilar', yetkazibberuvchiSchema);

```

Рисунок 2.6. Модель «Поставщики» в базе данных больницы

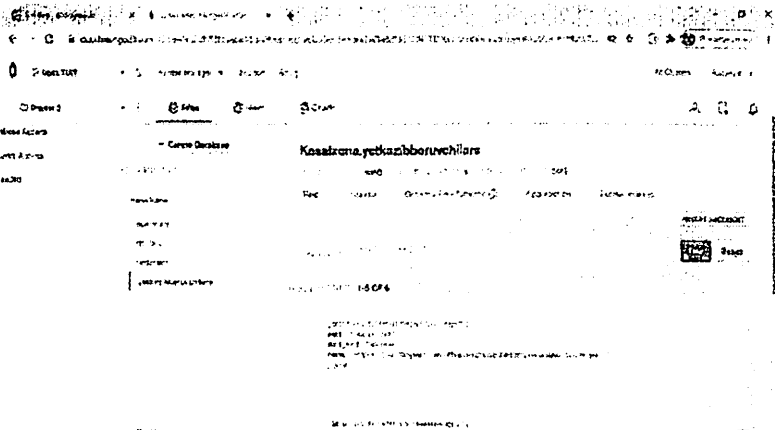


Рисунок 2.7. Множество «Поставщики» в базе данных больницы

После этого создадим множество «Лекарства» и создадим для него разделы «название», «рисунок», «тип», «цена», «количество» и «поставщик»:

```
one: don'tcheta = one Schema!!
  doos:
    type: String
    required: true
  yekazib beruvchi:
    type: mongoose.Schema.Types.ObjectId
    ref: "Yekazibovaru chilar"
  tarif:
    type: String
  rasud:
    type: String
  narxi:
    type: Number
    required: true
  miqdori:
    type: Number
```

Рисунок 2.8. Модель «Лекарства» в базе данных больницы

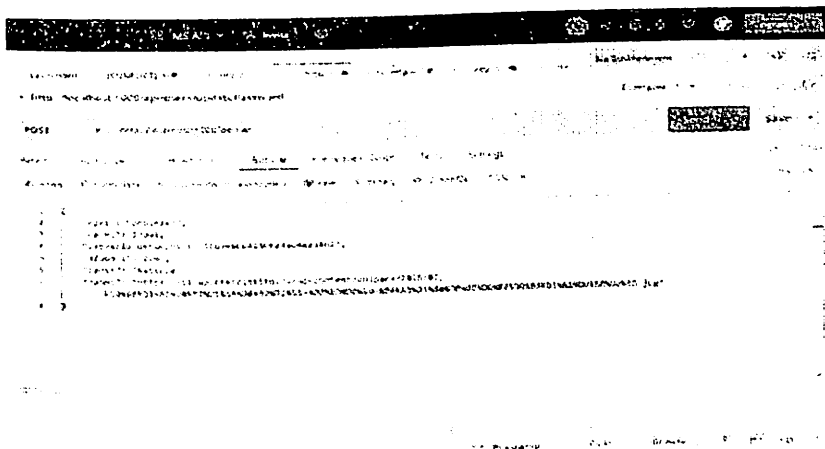


Рисунок 2.9. Добавление данных в множество «Лекарства»

И наконец, создаем множество «Пациенты» и включаем разделы «ФИО», «телефон», «возраст», «врач», «лекарства» и «адрес»:

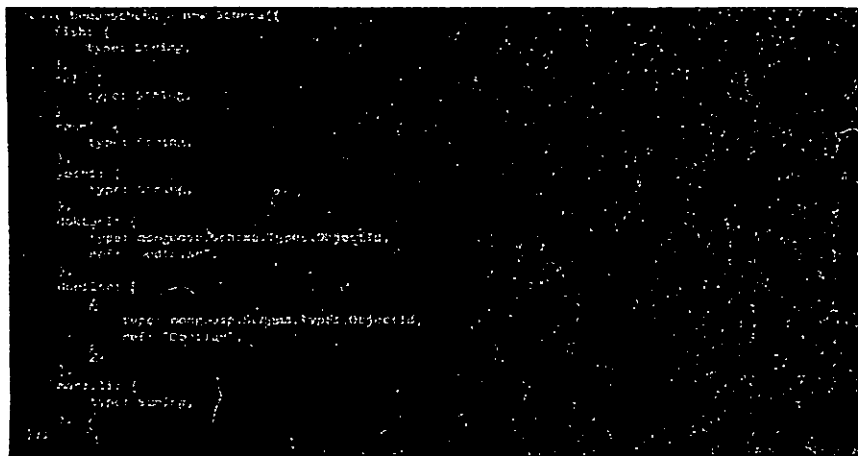


Рисунок 2.10. Модель «Пациент» в базе данных больницы

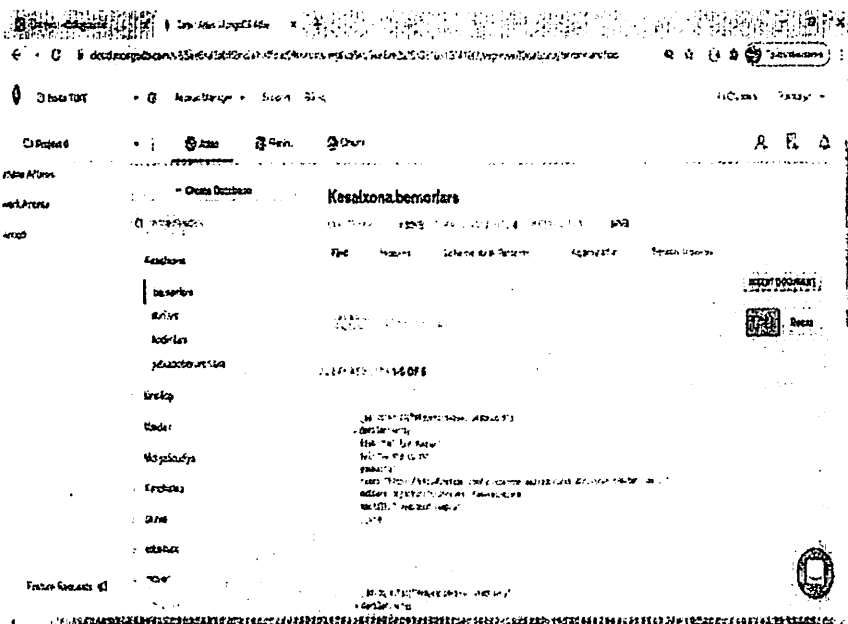


Рис. 2.11. Множество «Пациенты» в базе данных больницы

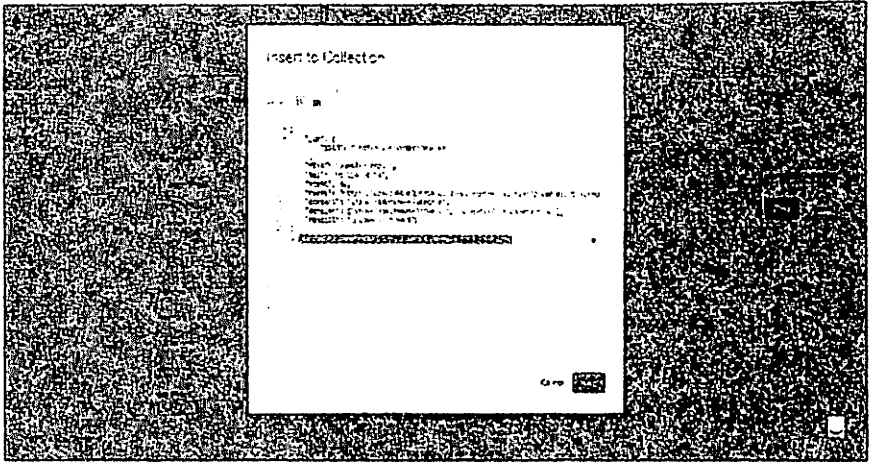


Рис. 2.12. Добавление данных в множество «Пациенты» через cloud.mongodb.com

2.3. Подключение базы данных MongoDB к Node.js

С помощью Node.js мы можем получать данные из персонального кабинета сайта cloud.mongodb.com и использовать там нашу базу данных в любом приложении. Ниже приведен код программы для подключения к базе данных «Больница», которую я создал в MongoDB с помощью Node.js:

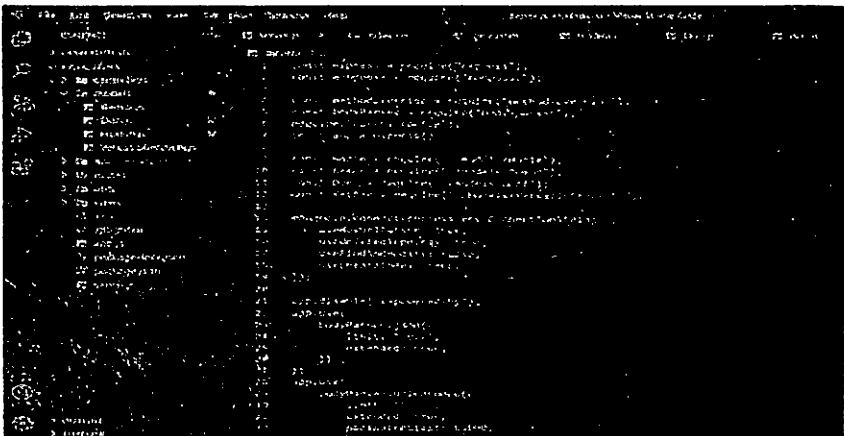


Рис. 2.13. Подключение к MongoDB через Node.js

Код подключения:

```
const express = require("express");
const mongoose = require("mongoose");
const methodOverride = require("method-override");
const bodyParser = require("body-parser");
require("dotenv/config");
const app = express();
const Hodim = require("./models/Hodim");
const Bemor = require("./models/Bemor");
const Dori = require("./models/Dori");
const YetBer = require("./models/YetkazibBeruvchi");
mongoose.connect(process.env.ConnectionString, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useFindAndModify: false,
  useCreateIndex: true,
});
app.disable("x-powered-by");
app.use(
  bodyParser.json({
    limit: "50mb",
    extended: true,
  })
);
app.use(
  bodyParser.urlencoded({
    limit: "50mb",
    extended: true,
    parameterLimit: 50000,
```



```

    })
  );
  app.use(function (req, res, next) {
    res.header("Access-Control-Allow-Origin", "*");
    res.header(
      "Access-Control-Allow-Headers",
      "Origin, X-Requested-With, Content-Type, Accept"
    );
    res.header("Access-Control-Expose-Headers", "x-token");
    next();
  });
  app.set("view engine", "ejs");
  app.use(express.urlencoded({ extended: false }));
  app.use(methodOverride("_method"));
  // asosiy routelar
  app.get("/", (req, res, next) => {
    res.render("index");
  });
  app.get("/hodimlar", async (req, res) => {
    const hodimlar = await Hodim.find().lean();
    res.render("hodimlar", { hodimlar: hodimlar });
  });
  app.post("/hodimlar", async (req, res, next) => {
    const hodim = await Hodim.create(req.body);
    res.status(200).json({
      success: true,
      data: xodim,
    });
  });
});

```

```

app.get("/bemorlar", async (req, res) => {
  const bemorlar = await Bemor.find()
    .populate([ { path: "doktori" }, { path: "dorilar" } ])
    .lean();
  res.render("bemorlar", { bemorlar: bemorlar });
});

app.post("/bemorlar", async (req, res, next) => {
  const bemor = await Bemor.create(req.body);
  res.status(200).json({
    success: true,
    data: bemor,
  });
});

app.get("/dorilar", async (req, res) => {
  const dorilar = await Dori.find().lean();
  res.render("dorilar", { dorilar: dorilar });
});

app.post("/dorilar", async (req, res, next) => {
  const dori = await Dori.create(req.body);
  res.status(200).json({
    success: true,
    data: dori,
  });
});

app.get("/yetberlar", async (req, res) => {
  const yetberlar = await YetBer.find().lean();
  res.render("yetberlar", { yetberlar: yetberlar });
});

app.post("/yetberlar", async (req, res, next) => {

```

```
const yetber = await YetBer.create(req.body);
res.status(200).json({
  success: true,
  data: yetber,
});
});
app.listen(5000);
```

Выводы по главе 2

В этой главе система базы данных больницы была создана и смоделирована в MongoDB, одной из систем управления базами данных NoSQL. Были описаны система базы данных больницы, процесс создания множеств и моделей и их структуры, инфологическая модель базы данных, связи между таблицами. Интерфейс программы реализован на языке программирования JavaScript, а подключение к базе данных реализовано на Node.js.

Система базы данных больницы была разработана в HTML и EJS (динамический HTML) с интерфейсом базы данных. В ходе этого процесса вышеуказанные технологии были детально изучены и внедрены на практике.

ГЛАВА 3. ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1. Хранилище данных и таблица данных системы базы данных больницы

Хранилище данных содержит всю информацию о системе баз данных больницы и их таблицах.

Информационная система – это система, которая должна заниматься вводом и контролем всех данных, а также их мониторингом. Собрана вся необходимая для системы информация. Я представил всю информацию в каждой отдельной таблице.

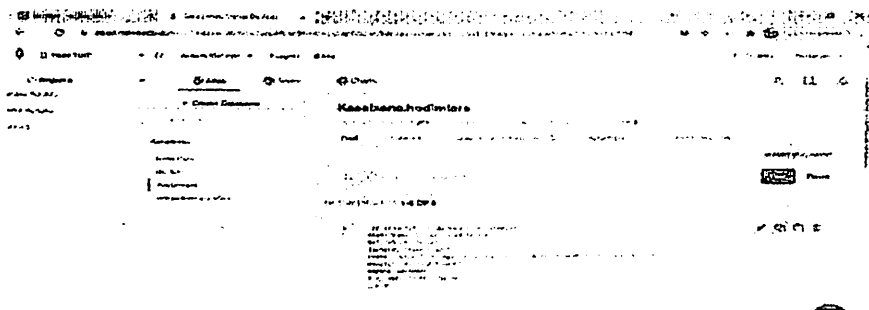


Рис. 3.2. Множество «Сотрудники» в базе данных больницы

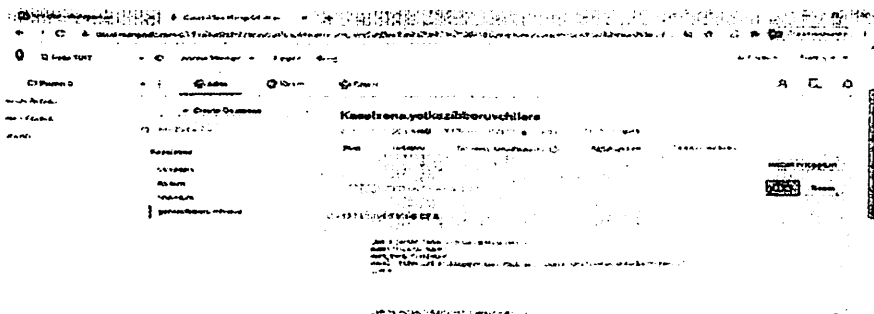


Рис. 3.2. Множество «Поставщики» в базе данных больницы

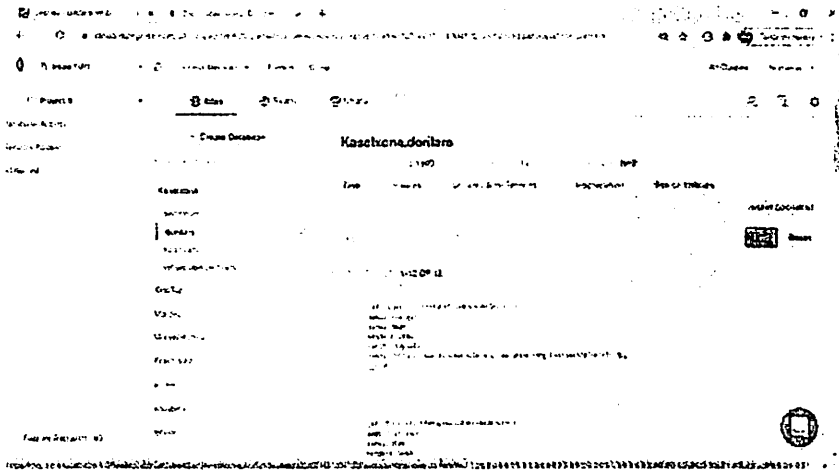


Рис. 3.3. Множество «Лекарства» в базе данных больницы

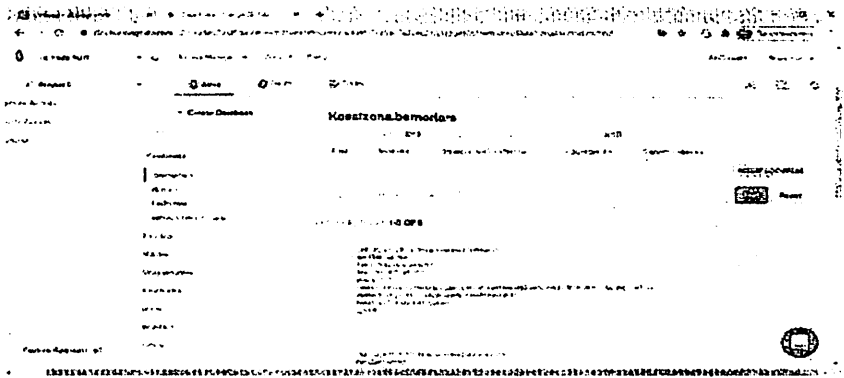


Рис. 3.4. Множество «Пациенты» в базе данных больницы

В среде программирования процесс создания программы определяется по поставленной задаче и реализуется по плану. Вопросы дизайна и базы данных также считаются наиболее важными аспектами.

В вышеприведенных таблицах не только постоянные пользователи, но и посетители сайта могут найти нужную им информацию. Эта информация

может быть обновлена Администратором через его персональный кабинет на сайте cloud.mongodb.com.

3.2. Интерфейс разработанного программного обеспечения

Информационная система – это система, которая должна заниматься вводом и контролем всех данных, а также их мониторингом. Собрана вся необходимая для системы информация. Меню ниже содержат все доступные параметры для персонала, а ниже находится домашняя страница системы баз данных больницы:



Рис. 3.5. Главная страницы системы «Больница»

Данная страница является главной страницей работы над проектом и базой данных больницы, отсюда пользователи могут легко переходить на другие страницы.

На следующей странице перечислены все сотрудники больницы, и здесь пациенты могут записаться на прием к врачу по своему выбору.

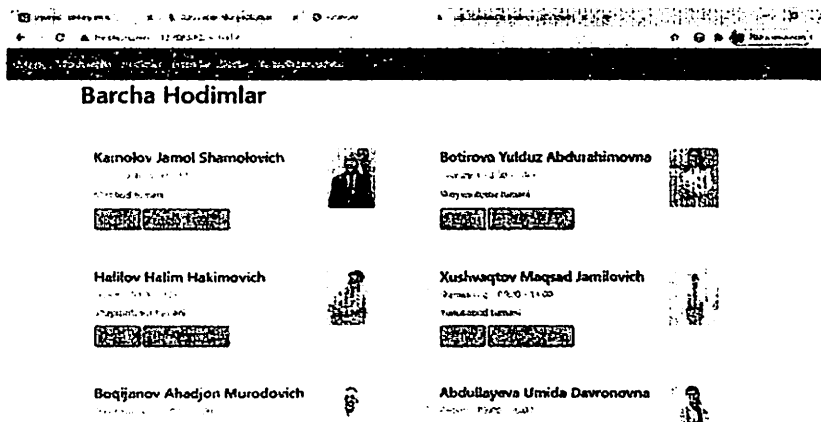


Рис.3.6. Страница «Все сотрудники» системы «Больница».

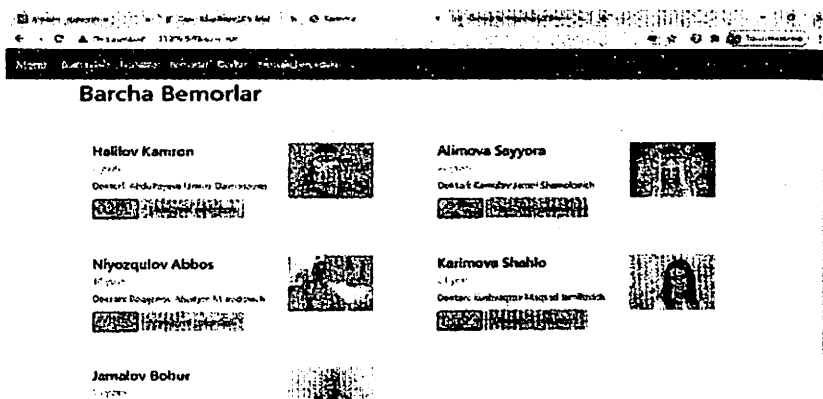


Рис. 3.7. Страница «Все пациенты» системы «Больница».

На странице выше перечислены все пациенты в больнице. Здесь врачи могут завершить процедуры для выздоровевших пациентов или получить подробную информацию о пациенте.

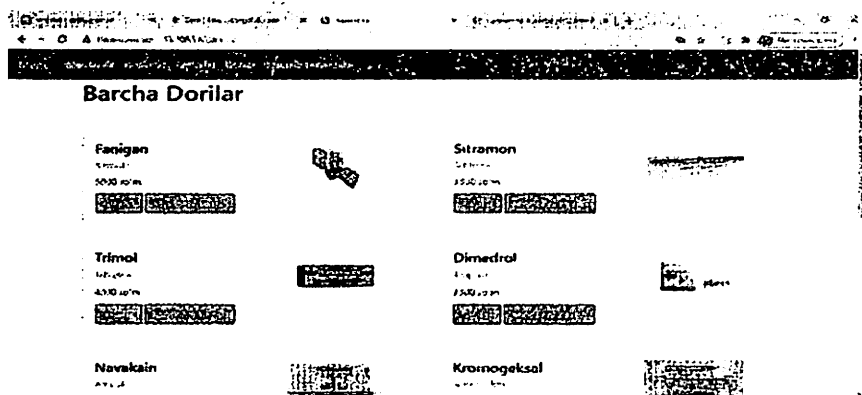


Рис. 3.8. Страница «Все препараты» системы «Больница».

На приведенной выше странице перечислены все лекарственные препараты, имеющиеся в больнице. С помощью этой страницы сотрудники могут получить подробную информацию о препарате или узнать, сколько его осталось.

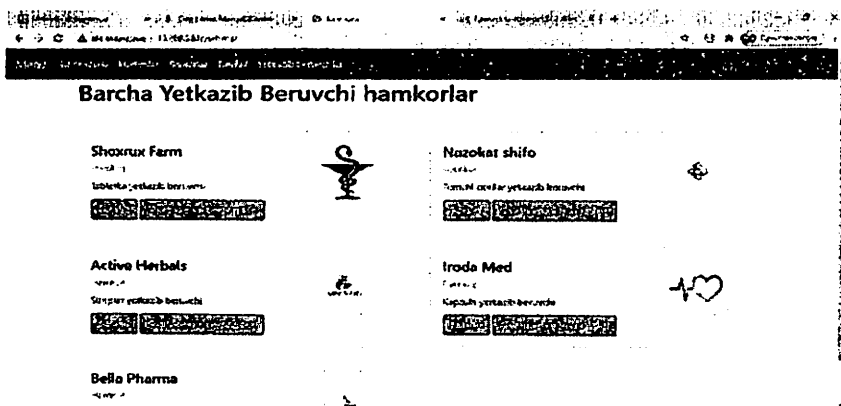


Рис. 3.9. Страница всех поставщиков лекарств системы «Больница».

На приведенной выше странице перечислены все существующие компании-партнеры по поставке лекарств, с которыми больницы заключили договора. На этой странице руководство может получить подробную информацию о партнере-поставщике или расторгнуть с ним существующий договор.

Выводы по главе 3

В этой главе была проведена работа по подключению базы данных, созданной в среде базы данных MongoDB, к среде Node.js. Для создания проекта выбрана определенная среда. Это: среды баз данных Node.js и MongoDB. В ходе работы над этим проектом было наработано много навыков работы со средой MongoDB. При выполнении поставленной задачи также возникло несколько проблем. Устранив их, автоматизированная система базы данных была доведена до завершения. Наиболее значительные из этих проблем возникли при подключении среды MongoDB к интерфейсу. Работа осуществлялась в среде программирования Node.js, в результате сформировались более прочные знания в этой области.

При создании базы данных в среде MongoDG были внесены различные изменения в некоторые процессы, то есть при соединении базы данных друг с другом, установлении связи между объектами, при создании нереляционной базы данных, и при работе с этой созданной базой данных и выработаны навыки работы в этой среде. Если возникнет необходимость работать в этих средах в будущем, появилась возможность использовать данные проекты и работать в этих средах. Я нашел для систему MongoDB совершенной базой данных. Я пришел к выводу, что обязательно должен использовать его в своих будущих проектах.

ЗАКЛЮЧЕНИЕ

На сегодняшний день в нашей стране проводится большая работа по развитию информационных технологий. Мы можем видеть отражение этого в постановлениях, законах и указах, принятых в этой сфере. Я приобрёл много знаний, работая над этим проектом.

Существует множество программ, которые выполняют вышеуказанные функции программы. Но данная программа имеет определенные преимущества.

В заключение можно сказать, что в ходе выполнения проектной работы были освоены ряд знаний. Работая над этим проектом, я приобрел много знаний и навыков. Уверен, что полученные знания и навыки помогут мне в будущем стать профессионалом в своем деле.

Сегодня в результате создания глобальных компьютерных сетей произошли кардинальные изменения в сфере передачи и распространения информации в мире. По данным международных аналитиков, около 30% всей корпоративной информации обменивается и хранится в электронном виде в базе данных, с разделением и без разделения по направлениям. Несомненно, в таких условиях электронный документооборот будет развиваться быстрее. Это значительно увеличивает скорость обмена информацией и значительно снижает потребность в пространстве для хранения.

В заключение следует сказать, что данная проектная работа призвана сэкономить время системы базы данных больницы, выполнить требования в короткие сроки и сократить чрезмерные расходы, а также облегчить работу сотрудников. В ходе этого проекта я получил много знаний и расширил имеющиеся у меня знания. Я научился работать с базой данных MongoDB и усовершенствовал свои навыки работы в среде Node.js.

Думаю, что на определенном уровне мне удалось реализовать намеченной цели. В целом, я надеюсь, что эта работа над проектом будет для

меня очень полезной и заложит основу для того, чтобы в будущем стать ведущим специалистом в своей профессии.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. O'zbekiston Respublikasi prezidentining 2017 yil 7 fevraldagi PF-4947-son
2. "O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida"gi Farmoni.
3. Mirziyoev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Toshkent. «O'zbekiston», NMIU, 2017. – 488 b.
4. В.П. Базы данных. Книга 2 распределенные и удаленные базы данных: учебник.// Москва ИД «ФОРУМ» - ИНФРА-М. – 2018. – С 261.
5. Rahul Batra. SQL Primer An Accelerated introduction to SQL Basics.// Gurgaon, India. 2019. – P 194.
6. Мартишин С.А. Базы данных. Практическое применение СУБД SQL – и NoSQL – типа для проектирования информационных систем: учеб. Пособие // - Москва: ИД «ФОРУМ» - ИНФРА-М, 2019, – 368 с.
7. Aripov M., Begalov B., Begumkulov U. va boshqalar. "Axborot texnologiyalari" / T.: "Noshir", 2009.-368 b.
8. Бегалов Б.А., Султанов С.М. "Информационные системы бухгалтерского учета": Проблемы, поиски, решения. Монография. – Т.: «Фан», 2009., 116 с.
9. Мусаев А.А., Бегалов Б.А. «Информационно - коммуникационные технологии в национальной экономике» Монография.-Т.: Фан, 2008.- 146 с.
10. Ayupov L.F., Begalov B.A. va boshqalar. "Shaxsiy kompyuterlar va ulardan samarali foydalanish asoslari". O'quv qo'llanma. TDIU.2007.
11. Informatika va axborot texnologiyalari . Akademik S.S.G'ulomov umumiy tahriri ostida. Darslik. "Iqtisodiyot" Toshkent. – 2009.
12. Informatika. Akademik S.S.G'ulomov umumiy tahriri ostida. Toshkent. – TDIU, 2007
13. Alimov R.X., Tulyaev D.R. Intellectual tizimlar. O'quv qo'llanma. TDIU.2010.

14. MongoDB architecture guide, 2018.
15. Node.js 8 – the Right Way, 2020.
16. Node.js in Action, 2020.
17. <https://www.mongodb.com>
18. <https://www.wikipedia.org>
19. <https://www.w3schools.com>
20. <https://www.nodejs.org>

ПРИЛОЖЕНИЕ

Dastur kodi: <https://github.com/MLRDer/kasalxona>

Dastur: <http://13.209.5.83/>

```
const express =
require("express");

    const mongoose = require("mongoose");
    const methodOverride = require("method-
override");
    const bodyParser = require("body-parser");
    require("dotenv/config");
    const app = express();
    const Hodim = require("./models/Hodim");
    const Bemor = require("./models/Bemor");
    const Dori = require("./models/Dori");
    const YetBer =
require("./models/YetkazibBeruvchi");
    mongoose.connect(process.env.ConnectionString, {
        useNewUrlParser: true,
        useUnifiedTopology: true,
        useFindAndModify: false,
        useCreateIndex: true,
    });
    app.disable("x-powered-by");
    app.use(
        bodyParser.json({
            limit: "50mb",
            extended: true,
        })
    )
```

```

    );
    app.use(
      bodyParser.urlencoded({
        limit: "50mb",
        extended: true,
        parameterLimit: 50000,
      })
    );
    app.use(function (req, res, next) {
      res.header("Access-Control-Allow-Origin",
        "*"); // update to match the domain you will make
        the request from
      res.header(
        "Access-Control-Allow-Headers",
const express = require("express");
const mongoose = require("mongoose");
const methodOverride = require("method-override");
const bodyParser = require("body-parser");
require("dotenv/config");
const app = express();
const Hodim = require("./models/Hodim");
const Bemor = require("./models/Bemor");
const Dori = require("./models/Dori");
const YetBer = require("./models/YetkazibBeruvchi");
mongoose.connect(process.env.ConnectionString, { useNewUrlParser:
true, useUnifiedTopology: true, useFindAndModify: false,
useCreateIndex: true, });
app.disable("x-powered-by");
      extended: true,

```

```

    })
  );
  app.use(
    bodyParser.urlencoded({
      limit: "50mb",
      extended: true,
      parameterLimit: 50000,
    })
  );
  app.use(function (req, res, next) {
    res.header("Access-Control-Allow-Origin", "*");
    // update to match the domain you will make the
    request from
    res.header(
      "Access-Control-Allow-Headers",
      const express = require("express");
      const mongoose = require("mongoose");
      const methodOverride = require("method-override");
      const bodyParser = require("body-parser");
require("dotenv/config");
const app = express();
const Hodim = require("./models/Hodim");
const Bemor = require("./models/Bemor");
const Dori = require("./models/Dori");
app.use( bodyParser.json({ limit: "50mb", extended: true, }) );
app.use( bodyParser.urlencoded({ limit: "50mb", extended: true,
parameterLimit: 50000, }) );
app.use(function (req, res, next) { res.header("Access-Control-
Allow-Origin", "*");

```



```

res.header( "Access-Control-Allow-Headers", "Origin, X-Requested-
With, Content-Type, Accept" );
res.header("Access-Control-Expose-Headers", "x-token");
next(); }); app.set("view engine", "ejs");
app.use(express.urlencoded({ extended: false }));
app.use(methodOverride("_method"));
// asosiy routelar app.get("/", (req, res, next) => {
res.render("index"); });
app.get("/hodimlar", async (req, res) => { const hodimlar =
await Hodim.find().lean();
res.render("hodimlar", { hodimlar: hodimlar }); });
app.post("/hodimlar", async (req, res, next) => { const hodim =
await Hodim.create(req.body);
res.status(200).json({ success: true, data: hodim, }); });
app.get("/bemorlar", async (req, res) => { const bemorlar = await
Bemor.find().populate([ { path: "doktori" }, { path: "dorilar"
} ]) .lean();
res.render("bemorlar", { bemorlar: bemorlar });
});
app.post("/bemorlar", async (req, res, next) => { const bemor =
await Bemor.create(req.body);
res.status(200).json({ success: true, data: bemor, }); });
app.get("/dorilar", async (req, res) => { const dorilar = await
Dori.find().lean();
res.render("dorilar", { dorilar: dorilar }); });
app.post("/dorilar", async (req, res, next) => { const dori =
await Dori.create(req.body);
res.status(200).json({ success: true, data: dori, }); });
app.get("/yetberlar", async (req, res) => { const yetberlar =

```

```
await YetBer.find().lean();
res.render("yetberlar", { yetberlar: yetberlar });
});
app.post("/yetberlar", async (req, res, next) => { const yetber =
await YetBer.create(req.body);
res.status(200).json({ success: true, data: yetber, }); });
app.listen(5000);
```

МЕТОДИЧЕСКИЙ УКАЗАНИЯ
для студентов направления бакалавриата
60610500 – «Компьютерный инжиниринг» («ИТ-сервис»)

Рассмотрено и рекомендовано к публикации на
заседании Научно-методического совета ТАТУ

Протокол № _____

“ _____ ” _____ 2022 год

Редактор: Маликова Н.Т., Джураев Т.Б.

Формат 60x84 1/16. Печ. лист 4,95.
Заказ № 124. Тираж 10.
Отпечатано в «Редакционно издательском»
отделе при ТУИТ.
Ташкент ул. Амир Темур, 108.