| Каркас   1. Прочитать задание: название, количество страниц (стиль, логотип и иконка) 2. Visual Studio WPF(Net Framework), название по заданию 3. Создали папки: ресурсы, pages, models, classes 4. Mainwindow: минимальные размеры, заголовок, столбцы/колонки (заливка), логотип, кнопки перехода, frame. название организации 5. App.xaml - стиль для кнопок 6. Класс manager: Frame Mainframe {get set} 7. Страница для пользователя и авторизации … (все по заданию) 8. Закрыли проект и загрузили на гит | Импорт   1. Посмотрите на названия файлов - это таблицы, которые нужно заполнить 2. Создайте новый файл Excel, чтобы не испортить данные в исходных файлах 3. Скопируйте заголовки таблиц из БД, чтобы понимать правильный порядок. Столбец ID оставляете пустым 4. Если каких-то данных нет, проверьте могут ли они быть нулевыми 5. Называйте листы именами таблиц, чтобы не путаться 6. При копировании использовать вставку значений 7. =ЕСЛИ(B2<>””;A2;C1) 8. =ЕСЛИ(B2<>””;A2;B2) 9. =ЕСЛИ(B2=”минут”;A2;A2\*60) - время\ 10. =J1&K1&L1 - Объединение ячеек 11. =J19&"."&K19&L19 - Объединение ячеек c добавлением точки 10 и 11 формулы нужны для объединение ячеек в которых указывается путь к файлу(папка, название файла, формат) |
| --- | --- |
| Вывод данных   1. В модели найти Context 2. Создать переменную и функцию Context 3. Разбить строки | |
| Вывод данных(DataGrid)   1. <DataGridTextColumn Header="id" Binding="{Binding id}" Width="90"/> | Вывод данных(ListView)   1. <TextBlock Text="{Binding login}"></TextBlock> |
| Добавление   1. Создание страницы edit и название страницы которую редактируем(EditWorkers) 2. Структура страницы:     TextBlock, TextBox, Button(F12 на click) расположение элементов или Grid или 2 StackPanel в WrapPanel  <TextBlock Margin="0 0 0 10">Время</TextBlock>  <TextBox Text="{Binding fio}" Margin="0 0 0 10"></TextBox>  3.2 Под IC пишем DataContext = currentworker;  3.3 При условии что ID = 0 через Add добавляем пользователя  if (currentworker.id == 0)  {  dozoEntities.GetContext().workers.Add(currentworker);  }  3.4 Сохраняем в базу и проверяем ошибки  try  {  using (var dbContextTransaction = dozoEntities.GetContext().Database.BeginTransaction())  {  try  {  if (currentworker.id == 0)  {  dozoEntities.GetContext().workers.Add(currentworker);  }  else  {  dozoEntities.GetContext().Entry(currentworker).State = EntityState.Modified;  }  dozoEntities.GetContext().SaveChanges();  dbContextTransaction.Commit();  MessageBox.Show("Всё сохранилось");  }  catch (Exception ex)  {  dbContextTransaction.Rollback();  MessageBox.Show($"Ошибка при сохранении изменений: {ex.Message}");  }  }  }  catch (Exception ex)  {  MessageBox.Show($"Общая ошибка: {ex.Message}");  }  Xaml.cs   1. Создать переменную для таблицы private workers currentworker = new workers();   Добавление времени  Вариант 1: Пользователь вводит время в ручную(дополнительное подключать календарь - <https://metanit.com/sharp/wpf/5.16.php> )  Вариант 2: Время берется из системы(  public DateTime datetoday = DateTime.Now;  currentworker.data = datetoday;  currentworker.time = datetoday.TimeOfDay;)  Фото:  public OpenFileDialog ofd = new OpenFileDialog();  private string newsourthpath = string.Empty;  string path = "";  private bool flag = false;    private void Foto(object sender, RoutedEventArgs e)  {  string Source = Environment.CurrentDirectory;  if (ofd.ShowDialog() == true)  {  flag = true;  string sourthpath = ofd.SafeFileName;  newsourthpath = Source.Replace("/bin/Debug", "/img/") + sourthpath;  PreviewImage.Source = new BitmapImage(new Uri(ofd.FileName));  path = ofd.FileName;  currentworker.photo = $"/img/{ofd.SafeFileName}";  }  } | Редактирование |