



**Министерство науки и высшего образования Российской
Федерации Федеральное государственное бюджетное
образовательное учреждение высшего образования «Московский
государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)» (МГТУ им.
Н.Э. Баумана)**

**Домашнее задание
по курсу «Разработка интернет-приложений»**

**Выполнил
студент группы ИУ5-54Б
Сысойкин Е.М.**

Москва, 2020

1 Описание задания

Стандартное задание:

Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

1. Две модели, связанные отношением один-ко-многим.
2. Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
3. Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
4. Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фреймворк по желанию студента.

Примечания:

1. Стандартный проект технопарка по Django приравнивается к стандартному варианту ДЗ.
2. По желанию студента для выполнения ДЗ может быть использован другой фреймворк, аналогичный Django, на любом языке программирования.

Расширенные задания, добавляющие +1 балл на экзамене:

1. Реализация домашнего задания с использованием фреймворка для разработки SPA-приложений (React, Angular, ...).
2. Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе) и развертывание приложения на облачном сервисе (например, heroku).
3. Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, <https://c3js.org/>) и развертывание приложения на облачном сервисе (например, heroku).
4. Подготовка черновика статьи по тематике курса (тематика статьи согласовывается с преподавателем).

2 Текст программы

main.go

```

1 package main
2
3 import (
4     "HW/handlers"
5     "HW/utils"
6     "log"
7     "net/http"
8 )
9
10 func main() {
11     conf := utils.ReadConfig()
12     applicationContext := utils.AppContext{
13         Config: conf,
14         DB:      utils.InitDatabase(conf),
15     }
16
17     http.HandleFunc("/", func(writer http.ResponseWriter, request *http.Request) {
18         log.Println(request.URL.Path)
19         handlers.MasterHandler(writer, request, &applicationContext)
20     })
21
22     http.HandleFunc("/detail", func(writer http.ResponseWriter, request *http.Request) {
23         log.Println(request.URL.Path)
24         handlers.DetailHandler(writer, request, &applicationContext)
25     })
26     http.HandleFunc("/stock", func(writer http.ResponseWriter, request *http.Request) {
27         log.Println(request.URL.Path)
28         handlers.StockHandler(writer, request, &applicationContext)
29     })
30     http.HandleFunc("/supplier", func(writer http.ResponseWriter, request *http.Request) {
31         log.Println(request.URL.Path)
32         handlers.SupplierHandler(writer, request, &applicationContext)
33     })
34
35     http.HandleFunc("/admin", func(writer http.ResponseWriter, request *http.Request) {
36         log.Println(request.URL.Path)
37         handlers.AdminHandler(writer, request, &applicationContext)
38     })
39
40     http.HandleFunc("/api/delete", func(writer http.ResponseWriter, request *http.Request)
41     ↪ {
42         log.Println(request.URL.Path)
43         handlers.DeleteHandler(writer, request, &applicationContext)
44     })

```

```

45 http.HandleFunc("/api/update", func(writer http.ResponseWriter, request *http.Request)
    ↪ {
46     log.Println(request.URL.Path)
47     handlers.UpdateHandler(writer, request, &applicationContext)
48 })
49
50 http.HandleFunc("/api/upload", func(writer http.ResponseWriter, request *http.Request)
    ↪ {
51     log.Println(request.URL.Path)
52     handlers.UploadImage(writer, request, &applicationContext)
53 })
54
55 http.HandleFunc("/api/remove", func(writer http.ResponseWriter, request *http.Request)
    ↪ {
56     log.Println(request.URL.Path)
57     handlers.DeleteFileHandler(writer, request, &applicationContext)
58 })
59
60 static := http.FileServer(http.Dir(conf.Root + "static"))
61 http.Handle("/static/", http.StripPrefix("/static/", static))
62
63 images := http.FileServer(http.Dir(conf.Root + "images"))
64 http.Handle("/img/", http.StripPrefix("/img/", images))
65
66 err := http.ListenAndServe(applicationContext.Config.Host, nil)
67 if err != nil {
68     log.Fatal(err)
69 }
70 }

```

utils/Databasesae.go

```

1 package utils
2
3 import (
4     "HW/models"
5     "gorm.io/driver/sqlite"
6     "gorm.io/gorm"
7     "log"
8 )
9
10 func InitDatabase(config Config) *gorm.DB {
11     db, err := gorm.Open(sqlite.Open(config.DatabasePath), &gorm.Config{})
12     if err != nil {
13         log.Fatalln("ERROR(failed to connect database): " + err.Error())
14     }

```

```

15
16     err = db.AutoMigrate(
17         &models.Detail{},
18         &models.Supplier{},
19         &models.DetailStock{},
20         &models.Stock{},
21     )
22     if err != nil {
23         log.Fatalln("ERROR(failed to migrate database): " + err.Error())
24     }
25     return db
26 }

```

utils/AppContext.go

```

1 package utils
2
3 import "gorm.io/gorm"
4
5 type AppContext struct {
6     Config Config
7     DB      *gorm.DB
8 }

```

utils/Configurator.go

```

1 package utils
2
3 import (
4     "github.com/BurntSushi/toml"
5     "io/ioutil"
6     "log"
7     "os"
8 )
9
10 type Config struct {
11     AdminToken string
12     Host        string
13     Root        string
14     DatabasePath string
15 }
16
17 var DefaultConfig = Config{
18     AdminToken: "admin",
19     Host:       "127.0.0.1:8080",
20     Root:       "",
21     DatabasePath: "hw.db",

```

```

22 }
23
24 var defConfigLocation = "./hw.conf"
25
26 func ReadConfig() Config {
27     var configLocation string
28     if len(os.Args) > 1 {
29         configLocation = os.Args[1]
30     } else {
31         configLocation = defConfigLocation
32     }
33     configBytes, err := ioutil.ReadFile(configLocation)
34     if err != nil {
35         log.Println("ERROR: Failed to read config file: " + err.Error())
36         log.Println("Warning: use default config instead")
37         return DefaultConfig
38     }
39     tomlData := string(configBytes)
40     var config = DefaultConfig
41     if _, err := toml.Decode(tomlData, &config); err != nil {
42         log.Println("ERROR: Failed to parse config file: " + err.Error())
43         log.Println("Warning: use default config instead")
44         return DefaultConfig
45     }
46     log.Println("Root: " + config.Root)
47     log.Println("Host: " + config.Host)
48     log.Println("Database: " + config.DatabasePath)
49     log.Println("Token: " + config.AdminToken)
50     return config
51 }

```

handlers/Base.go

```

1 package handlers
2
3 import (
4     "HW/utils"
5     "fmt"
6     "html/template"
7     "log"
8     "net/http"
9 )
10
11 type ErrorStruct struct {
12     Code      int
13     Description string

```

```

14 }
15
16 var HttpNotFound = ErrorStruct{Code: 404, Description: "Page not found"}
17 var HttpForbidden = ErrorStruct{Code: 403, Description: "Access denied"}
18 var HttpInternalServerError = ErrorStruct{Code: 500, Description: "Internal server error"}
19
20
21 func writeCode(error ErrorStruct, w http.ResponseWriter, ctx *utils.AppContext) {
22     tpl, err := template.ParseFiles(ctx.Config.Root + "templates/error.html")
23     if err != nil {
24         log.Println(err)
25         return
26     }
27     err = tpl.Execute(w, error)
28     if err != nil {
29         log.Println(err)
30     }
31 }
32
33 func writeHttpCode(code int, w http.ResponseWriter) {
34     log.Println(fmt.Sprintf("Error(code): %d", code))
35     w.WriteHeader(code)
36     switch code {
37     case http.StatusForbidden:
38         _, _ = w.Write([]byte("403 Access denied."))
39     case http.StatusInternalServerError:
40         _, _ = w.Write([]byte("500 Internal server error."))
41     case http.StatusNotFound:
42         _, _ = w.Write([]byte("404 Not found."))
43     case http.StatusBadRequest:
44         _, _ = w.Write([]byte("400 Bad request. Check params."))
45     }
46 }

```

handlers/AdminApi.go

```

1 package handlers
2
3 import (
4     "HW/models"
5     "HW/utils"
6     "crypto/sha1"
7     "encoding/base64"
8     "encoding/json"
9     "fmt"
10    "io/ioutil"

```

```

11     "log"
12     "net/http"
13     "os"
14     "strconv"
15 )
16
17 func DeleteHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
18     token := r.URL.Query().Get("token")
19     if token != ctx.Config.AdminToken {
20         w.WriteHeader(http.StatusForbidden)
21         return
22     }
23
24     tableStr := r.URL.Query().Get("table")
25     idStr := r.URL.Query().Get("id")
26     id, _ := strconv.Atoi(idStr)
27
28     switch tableStr {
29     case "details":
30         ctx.DB.Unscoped().Delete(&models.Detail{}, id)
31     case "suppliers":
32         ctx.DB.Unscoped().Delete(&models.Supplier{}, id)
33     case "stocks":
34         ctx.DB.Unscoped().Delete(&models.Stock{}, id)
35     case "detail_stocks":
36         ctx.DB.Unscoped().Delete(&models.DetailStock{}, id)
37     }
38 }
39
40 func UpdateHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
41     token := r.URL.Query().Get("token")
42     if token != ctx.Config.AdminToken {
43         writeHttpCode(http.StatusForbidden, w)
44         return
45     }
46
47     tableStr := r.URL.Query().Get("table")
48
49     bodyBytes, err := ioutil.ReadAll(r.Body)
50     if err != nil {
51         log.Println(err)
52         writeHttpCode(http.StatusInternalServerError, w)
53         return
54     }
55

```



```

56     switch tableStr {
57     case "details":
58         var detail = models.Detail{}
59         err := json.Unmarshal(bodyBytes, &detail)
60         if err != nil {
61             log.Println(err)
62             writeHttpCode(http.StatusBadRequest, w)
63             return
64         }
65         ctx.DB.Save(&detail)
66     case "suppliers":
67         var supplier = models.Supplier{}
68         err := json.Unmarshal(bodyBytes, &supplier)
69         if err != nil {
70             log.Println(err)
71             writeHttpCode(http.StatusBadRequest, w)
72             return
73         }
74         ctx.DB.Save(&supplier)
75     case "stocks":
76         var stock = models.Stock{}
77         err := json.Unmarshal(bodyBytes, &stock)
78         if err != nil {
79             log.Println(err)
80             writeHttpCode(http.StatusBadRequest, w)
81             return
82         }
83         ctx.DB.Save(&stock)
84     case "detail_stocks":
85         var detailStock = models.DetailStock{}
86         err := json.Unmarshal(bodyBytes, &detailStock)
87         if err != nil {
88             log.Println(err)
89             writeHttpCode(http.StatusBadRequest, w)
90             return
91         }
92         ctx.DB.Save(&detailStock)
93     }
94 }
95
96
97 func UploadImage(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
98     token := r.URL.Query().Get("token")
99     if token != ctx.Config.AdminToken {
100         writeHttpCode(http.StatusForbidden, w)

```

```

101     return
102 }
103
104 err := r.ParseMultipartForm(5 < 20)
105 if err != nil {
106     log.Println(err)
107     writeHttpCode(http.StatusInternalServerError, w)
108     return
109 }
110 file, handler, err := r.FormFile("file")
111 if err != nil {
112     log.Println(err)
113     writeHttpCode(http.StatusBadRequest, w)
114     return
115 }
116 defer file.Close()
117 fmt.Printf("Uploaded File: %v\n", handler.Filename)
118 fmt.Printf("File Size: %v\n", handler.Size)
119 fmt.Printf("MIME Header: %v\n", handler.Header)
120
121 fileBytes, err := ioutil.ReadAll(file)
122 if err != nil {
123     log.Println(err)
124     writeHttpCode(http.StatusInternalServerError, w)
125     return
126 }
127
128 hasher := sha1.New()
129 hasher.Write(fileBytes)
130 fileId := base64.URLEncoding.EncodeToString(hasher.Sum(nil))
131 err = ioutil.WriteFile(ctx.Config.Root+"images/"+fileId, fileBytes, 0644)
132 if err != nil {
133     log.Println(err)
134     writeHttpCode(http.StatusInternalServerError, w)
135     return
136 }
137 }
138
139 func DeleteFileHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
140     token := r.URL.Query().Get("token")
141     if token != ctx.Config.AdminToken {
142         writeHttpCode(http.StatusForbidden, w)
143         return
144     }
145     fileId := r.URL.Query().Get("fileID")

```

```

146     if len(fileID) == 0 {
147         writeHttpCode(http.StatusBadRequest, w)
148         return
149     }
150     err := os.Remove(ctx.Config.Root + "images/" + fileID)
151     if err != nil {
152         writeHttpCode(http.StatusInternalServerError, w)
153         return
154     }
155 }
156 }

```

handlers/AdminHandler.go

```

1  package handlers
2
3  import (
4      "HW/models"
5      "HW/utils"
6      "html/template"
7      "io/ioutil"
8      "log"
9      "net/http"
10     "strconv"
11 )
12
13 type AdminHtmlPageObject struct {
14     UrlParams    []UrlParam
15     Details      []*models.Detail
16     Suppliers    []*models.Supplier
17     Stocks       []*models.Stock
18     DetailStocks []*models.DetailStock
19     ImagesIDs    []*string
20 }
21
22 type UrlParam struct {
23     Token  string
24     Table  string
25     Label  string
26     Offset string
27 }
28
29 const PAGELIMIT = 15
30
31 func AdminHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
32     w.Header().Set("Content-Type", "text/html")

```

```

33 token := r.URL.Query().Get("token")
34 if token != ctx.Config.AdminToken {
35     writeCode(HttpForbidden, w, ctx)
36     return
37 }
38 tableStr := r.URL.Query().Get("table")
39 offsetStr := r.URL.Query().Get("offset")
40
41 var offset, _ = strconv.Atoi(offsetStr)
42
43 var pageData = AdminHtmlPageObject{
44     UrlParams: []UrlParam{
45         {Token: token, Table: "details", Label: "Детали", Offset: offsetStr},
46         {Token: token, Table: "suppliers", Label: "Поставщики", Offset: offsetStr},
47         {Token: token, Table: "stocks", Label: "Склады", Offset: offsetStr},
48         {Token: token, Table: "detail_stocks", Label: "Детали на складах", Offset:
49             ↪ offsetStr},
50         {Token: token, Table: "images", Label: "Изображения", Offset: offsetStr},
51     },
52 }
53
54 switch tableStr {
55 case "details":
56     pageData.Details = &[]models.Detail{}
57     ctx.DB.Offset(offset).Limit(PAGELIMIT).Find(pageData.Details)
58 case "suppliers":
59     pageData.Suppliers = &[]models.Supplier{}
60     ctx.DB.Offset(offset).Limit(PAGELIMIT).Find(pageData.Suppliers)
61 case "stocks":
62     pageData.Stocks = &[]models.Stock{}
63     ctx.DB.Offset(offset).Limit(PAGELIMIT).Find(pageData.Stocks)
64 case "detail_stocks":
65     pageData.DetailStocks = &[]models.DetailStock{}
66     ctx.DB.Offset(offset).Limit(PAGELIMIT).Find(pageData.DetailStocks)
67 case "images":
68     files, err := ioutil.ReadDir(ctx.Config.Root + "images")
69     if err != nil {
70         writeCode(HttpInternalServerError, w, ctx)
71         return
72     }
73
74     pageData.ImagesIDs = &[]string{}
75     for _, file := range files {
76         if !file.IsDir() {
77             *pageData.ImagesIDs = append(*pageData.ImagesIDs, file.Name())
78         }
79     }
80 }

```

```

77         }
78     }
79 }
80
81 tmpl, err := template.ParseFiles(ctx.Config.Root + "templates/admin.html")
82 if err != nil {
83     log.Println(err)
84     return
85 }
86 err = tmpl.Execute(w, pageData)
87 if err != nil {
88     log.Println(err)
89     return
90 }
91
92 }

```

handlers/StockHandler.go

```

1  package handlers
2
3  import (
4      "HW/models"
5      "HW/utils"
6      "html/template"
7      "log"
8      "net/http"
9      "strconv"
10 )
11
12 type StockHtmlPageObject struct {
13     Stock      models.Stock
14     Details    []*models.DetailView
15     Suppliers  []*models.Supplier
16 }
17
18 func StockHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
19     w.Header().Set("Content-Type", "text/html")
20
21     idStr := r.URL.Query().Get("id")
22     if len(idStr) == 0 {
23         writeCode(HttpNotFound, w, ctx)
24         return
25     }
26     id, err := strconv.Atoi(idStr)
27     if err != nil {

```

```

28     writeCode(InternalServerError, w, ctx)
29     return
30 }
31
32 var pageData StockHtmlPageObject
33 query := ctx.DB.Find(&pageData.Stock, id)
34 if query.RowsAffected == 0 {
35     writeCode(NotFound, w, ctx)
36     return
37 }
38
39 var details []models.DetailView
40 ctx.DB.
41     Select("`details`.*, `detail_stocks`.count, `suppliers`.name as supplier_name").
42     Model(&models.Detail{}).
43     Joins("left join suppliers on suppliers.id = details.supplier").
44     Joins("left join detail_stocks on details.id = detail_stocks.detail_id").
45     Joins("left join stocks on stocks.id = detail_stocks.stock_id").
46     Where("stocks.id = ?", pageData.Stock.ID).
47     Limit(1000).
48     Scan(&details)
49
50 if len(details) != 0 {
51     pageData.Details = &details
52 }
53
54 tmpl, err := template.ParseFiles(ctx.Config.Root + "templates/stock.html")
55 if err != nil {
56     log.Println(err)
57     return
58 }
59 err = tmpl.Execute(w, pageData)
60 if err != nil {
61     log.Println(err)
62     return
63 }
64
65 }

```

handlers/DetailHandler.go

```

1 package handlers
2
3 import (
4     "HW/models"
5     "HW/utis"

```

```

6     "html/template"
7     "log"
8     "net/http"
9     "strconv"
10 )
11
12 type DetailHtmlPageObject struct {
13     Detail    models.Detail
14     Supplier  *models.Supplier
15     Stocks    []*models.StockView
16 }
17
18 func DetailHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
19     w.Header().Set("Content-Type", "text/html")
20
21     idStr := r.URL.Query().Get("id")
22     if len(idStr) == 0 {
23         writeCode(HttpNotFound, w, ctx)
24         return
25     }
26     id, err := strconv.Atoi(idStr)
27     if err != nil {
28         writeCode(HttpInternalServerError, w, ctx)
29         return
30     }
31
32     var pageData DetailHtmlPageObject
33     query := ctx.DB.Find(&pageData.Detail, id)
34     if query.RowsAffected == 0 {
35         writeCode(HttpNotFound, w, ctx)
36         return
37     }
38
39     pageData.Supplier = &models.Supplier{}
40     query = ctx.DB.Find(pageData.Supplier, pageData.Detail.Supplier)
41     if query.RowsAffected == 0 {
42         pageData.Supplier = nil
43     }
44
45     var stocks []*models.StockView
46     ctx.DB.
47         Select("`stocks`.*, `detail_stocks`.count").
48         Model(&models.Stock{}).
49         Joins("left join detail_stocks on stocks.id = detail_stocks.stock_id").
50         Where("detail_stocks.detail_id = ?", pageData.Detail.ID).

```

```

51         Limit(1000).
52         Scan(&stocks)
53     if len(stocks) != 0 {
54         pageData.Stocks = &stocks
55     }
56
57     tmpl, err := template.ParseFiles(ctx.Config.Root + "templates/detail.html")
58     if err != nil {
59         log.Println(err)
60         return
61     }
62     err = tmpl.Execute(w, pageData)
63     if err != nil {
64         log.Println(err)
65         return
66     }
67
68 }

```

handlers/MasterHandler.go

```

1  package handlers
2
3  import (
4      "HW/models"
5      "HW/utils"
6      "html/template"
7      "log"
8      "net/http"
9  )
10
11  type MasterHtmlPageObject struct {
12      Details []models.Detail
13      Suppliers []models.Supplier
14      Stocks []models.Stock
15  }
16
17  func MasterHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
18      w.Header().Set("Content-Type", "text/html")
19      tmpl, err := template.ParseFiles(ctx.Config.Root + "templates/master.html")
20      if err != nil {
21          log.Println(err)
22          writeCode(HttpInternalServerError, w, ctx)
23          return
24      }
25      var pageData MasterHtmlPageObject

```



```

26     ctx.DB.Find(&pageData.Details).Limit(1000)
27     ctx.DB.Find(&pageData.Suppliers).Limit(1000)
28     ctx.DB.Find(&pageData.Stocks).Limit(1000)
29
30     err = tpl.Execute(w, pageData)
31     if err != nil {
32         log.Println(err)
33         writeCode(InternalServerError, w, ctx)
34         return
35     }
36 }

```

handlers/SupplierHandler.go

```

1  package handlers
2
3  import (
4      "HW/models"
5      "HW/utils"
6      "html/template"
7      "log"
8      "net/http"
9      "strconv"
10 )
11
12 type SupplierHtmlPageObject struct {
13     Supplier models.Supplier
14     Details  *[]models.Detail
15     Stocks   *[]models.Stock
16 }
17
18 func SupplierHandler(w http.ResponseWriter, r *http.Request, ctx *utils.AppContext) {
19     w.Header().Set("Content-Type", "text/html")
20
21     idStr := r.URL.Query().Get("id")
22     if len(idStr) == 0 {
23         writeCode(NotFound, w, ctx)
24         return
25     }
26     id, err := strconv.Atoi(idStr)
27     if err != nil {
28         writeCode(InternalServerError, w, ctx)
29         return
30     }
31
32     var pageData SupplierHtmlPageObject

```

```

33 query := ctx.DB.Find(&pageData.Supplier, id)
34 if query.RowsAffected == 0 {
35     writeCode(HttpNotFound, w, ctx)
36     return
37 }
38
39 var details []models.Detail
40 ctx.DB.Find(&details).Where("supplier = ?", pageData.Supplier.ID)
41 if len(details) != 0 {
42     pageData.Details = &details
43 }
44
45 var stocks []models.Stock
46 ctx.DB.
47     Model(&models.Stock{}).
48     Distinct().
49     Joins("left join detail_stocks on stocks.id = detail_stocks.stock_id").
50     Joins("left join details on details.id = detail_stocks.detail_id").
51     Where("details.supplier = ?", pageData.Supplier.ID).
52     Limit(1000).
53     Scan(&stocks)
54
55 if len(stocks) != 0 {
56     pageData.Stocks = &stocks
57 }
58
59 tmpl, err := template.ParseFiles(ctx.Config.Root + "templates/supplier.html")
60 if err != nil {
61     log.Println(err)
62     return
63 }
64 err = tmpl.Execute(w, pageData)
65 if err != nil {
66     log.Println(err)
67     return
68 }
69
70 }

```

templates/admin.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Admin</title>

```

```

6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link href="/static/css/bootstrap.min.css" rel="stylesheet">
8 <link href="/static/css/all.min.css" rel="stylesheet"> <!-- Font Awesome -->
9 <script src="/static/js/bootstrap.bundle.min.js"></script>
10
11 <style>
12     .limited-keepaspect {
13         max-width: 256px;
14         max-height: 256px;
15         width: auto;
16         height: auto;
17     }
18 </style>
19
20 <script>
21     function editStock(id) {
22         document.getElementById("stock_id").value = document.getElementById("stock_id_"
23         ↪ + id).innerText;
24         document.getElementById("stock_addr").value =
25         ↪ document.getElementById("stock_addr_" + id).innerText;
26         document.getElementById("stock_phone").value =
27         ↪ document.getElementById("stock_phone_" + id).innerText;
28     }
29
30     function newStock() {
31         document.getElementById("stock_id").value = 0;
32         document.getElementById("stock_addr").value = "";
33         document.getElementById("stock_phone").value = "";
34     }
35
36     async function deleteRow(id, table) {
37         let response = await fetch(window.location.protocol + '//' +
38         ↪ window.location.host + '/api/delete?token=' + getToken() + '&table=' +
39         ↪ table + '&id=' + id);
40         if (response.ok) {
41             alert('OK!');
42             location.reload();
43         }
44     }
45
46     async function updateNewStock() {
47         let response = await fetch(window.location.protocol + '//' +
48         ↪ window.location.host + '/api/update?token=' + getToken() + '&table=stocks',
49         ↪ {
50             method: 'POST',

```

```

44     headers: {
45         'Content-Type': 'application/json;charset=utf-8'
46     },
47     body: JSON.stringify({
48         ID: Number(document.getElementById("stock_id").value),
49         Address: document.getElementById("stock_addr").value,
50         Phone: document.getElementById("stock_phone").value
51     })
52 });
53 if (response.ok) {
54     alert('OK!');
55     location.reload();
56 }
57 }
58
59
60 function getToken() {
61     const urlParams = new URLSearchParams(window.location.search);
62     return urlParams.get('token');
63 }
64
65 async function updateNewDetail() {
66     let response = await fetch(window.location.protocol + '//' +
67     ↪ window.location.host + '/api/update?token=' + getToken() +
68     ↪ '&table=details', {
69         method: 'POST',
70         headers: {
71             'Content-Type': 'application/json;charset=utf-8'
72         },
73         body: JSON.stringify({
74             ID: Number(document.getElementById("detail_id").value),
75             Name: document.getElementById("detail_name").value,
76             Description: document.getElementById("detail_description").value,
77             Weight: Number(document.getElementById("detail_weight").value),
78             Supplier: Number(document.getElementById("detail_supplier").value),
79             Image: document.getElementById("detail_image").value
80         })
81     });
82     if (response.ok) {
83         alert('OK!');
84         location.reload();
85     }
86 }
87
88 function editDetail(id) {

```

```

87     document.getElementById("detail_id").value =
      ↳ document.getElementById("detail_id_" + id).innerText;
88     document.getElementById("detail_name").value =
      ↳ document.getElementById("detail_name_" + id).innerText;
89     document.getElementById("detail_description").value =
      ↳ document.getElementById("detail_description_" + id).innerText;
90     document.getElementById("detail_weight").value =
      ↳ document.getElementById("detail_weight_" + id).innerText;
91     document.getElementById("detail_supplier").value =
      ↳ document.getElementById("detail_supplier_" + id).innerText;
92     document.getElementById("detail_image").value =
      ↳ document.getElementById("detail_image_" + id).innerText;
93 }
94
95 function newDetail() {
96     document.getElementById("detail_id").value = 0;
97     document.getElementById("detail_name").value = "";
98     document.getElementById("detail_description").value = "";
99     document.getElementById("detail_weight").value = "";
100    document.getElementById("detail_supplier").value = "";
101    document.getElementById("detail_image").value = "";
102 }
103
104 async function updateNewSupplier() {
105     let response = await fetch(window.location.protocol + '//' +
      ↳ window.location.host + '/api/update?token=' + getToken() +
      ↳ '&table=suppliers', {
106         method: 'POST',
107         headers: {
108             'Content-Type': 'application/json;charset=utf-8'
109         },
110         body: JSON.stringify({
111             ID: Number(document.getElementById("supplier_id").value),
112             Name: document.getElementById("supplier_name").value,
113             Address: document.getElementById("supplier_addr").value,
114             Phone: document.getElementById("supplier_phone").value
115         })
116     });
117     if (response.ok) {
118         alert('OK!');
119         location.reload();
120     }
121
122 }
123

```

```

124 function editSupplier(id) {
125     document.getElementById("supplier_id").value =
        ↳ document.getElementById("supplier_id_" + id).innerText;
126     document.getElementById("supplier_name").value =
        ↳ document.getElementById("supplier_name_" + id).innerText;
127     document.getElementById("supplier_addr").value =
        ↳ document.getElementById("supplier_addr_" + id).innerText;
128     document.getElementById("supplier_phone").value =
        ↳ document.getElementById("supplier_phone_" + id).innerText;
129 }

130
131 function newSupplier() {
132     document.getElementById("supplier_id").value = 0;
133     document.getElementById("supplier_name").value = "";
134     document.getElementById("supplier_addr").value = "";
135     document.getElementById("supplier_phone").value = "";
136 }

137
138 async function updateNewDetailStock() {
139     let response = await fetch(window.location.protocol + '//' +
        ↳ window.location.host + '/api/update?token=' + getToken() +
        ↳ '&table=detail_stocks', {
140         method: 'POST',
141         headers: {
142             'Content-Type': 'application/json;charset=utf-8'
143         },
144         body: JSON.stringify({
145             ID: Number(document.getElementById("detail_stock_id").value),
146             StockID:
                ↳ Number(document.getElementById("detail_stock_stock_id").value),
147             DetailID:
                ↳ Number(document.getElementById("detail_stock_detail_id").value),
148             Count: Number(document.getElementById("detail_stock_count").value)
149         })
150     });
151     if (response.ok) {
152         alert('OK!');
153         location.reload();
154     }
155
156 }

157
158 function editDetailStock(id) {
159     document.getElementById("detail_stock_id").value =
        ↳ document.getElementById("detail_stock_id_" + id).innerText;

```

```

160     document.getElementById("detail_stock_detail_id").value =
        ↳ document.getElementById("detail_stock_detail_id_" + id).innerText;
161     document.getElementById("detail_stock_stock_id").value =
        ↳ document.getElementById("detail_stock_stock_id_" + id).innerText;
162     document.getElementById("detail_stock_count").value =
        ↳ document.getElementById("detail_stock_count_" + id).innerText;
163 }
164
165 function newDetailStock() {
166     document.getElementById("detail_stock_detail_id").value = "";
167     document.getElementById("detail_stock_stock_id").value = "";
168     document.getElementById("detail_stock_count").value = "";
169 }
170
171 async function uploadImage() {
172     let photo = document.getElementById("img").files[0];
173     let formData = new FormData();
174
175     formData.append("file", photo);
176     let response = await fetch(window.location.protocol + '//' +
        ↳ window.location.host + '/api/upload?token=' + getToken(), {
177         method: "POST",
178         body: formData
179     });
180     if (response.ok) {
181         alert('OK!');
182         location.reload();
183     }
184 }
185
186 async function deleteImage(fileID) {
187     let response = await fetch(window.location.protocol + '//' +
        ↳ window.location.host + '/api/remove?token=' + getToken() + '&fileID=' +
        ↳ fileID);
188     if (response.ok) {
189         alert('OK!');
190         location.reload();
191     }
192 }
193 </script>
194 </head>
195 <body>
196 <nav class="navbar navbar-expand navbar-dark bg-dark">
197     <div class="container-fluid">
198         <h2 class="navbar-brand">Admin</h2>

```

```

199     <div class="navbar-collapse" id="navbarSupportedContent">
200         <ul class="navbar-nav me-auto mb-2">
201             <li class="nav-item">
202                 <a class="nav-link" href="/">Главная</a>
203             </li>
204             {{ range .UrlParams }}
205                 <li class="nav-item">
206                     <a class="nav-link"
207                         href="/admin?token={{ .Token }}&table={{ .Table }}&offset={{
208                             ↪ .Offset }}">{{ .Label }}</a>
209                 </li>
210             {{ end }}
211         </ul>
212     </div>
213 </nav>
214
215 {{ if .Details }}
216     <div class="m-5">
217         <h1>Детали:</h1>
218         <div class="row align-items-start">
219
220             <table class="table table-striped col">
221                 <thead>
222                     <tr>
223                         <th scope="col">
224                             <a class="btn btn-primary btn-sm" role="button"
225                                 ↪ onclick="newDetail()"><i
226                                     class="fas fa-plus"></i></a>
227                         </th>
228                         <th scope="col">Номер</th>
229                         <th scope="col">Имя</th>
230                         <th scope="col">Описание</th>
231                         <th scope="col">Вес</th>
232                         <th scope="col">Превью</th>
233                         <th scope="col">Поставщик</th>
234                     </tr>
235                 </thead>
236                 <tbody>
237                     {{ range .Details }}
238                         <tr>
239                             <td>
240                                 <div class="btn-group" role="group">
241                                     <a class="btn btn-warning btn-sm" role="button"
242                                         ↪ onclick="editDetail({{ .ID }})"><i

```



```

241         class="fas fa-edit"></i> </a>
242     <a class="btn btn-danger btn-sm" role="button"
243         onclick="deleteRow({{ .ID }}, 'details')"><i class="fas
244         ↪ fa-trash"></i></a>
245     </div>
246 </td>
247 <td id="detail_id_{{ .ID }}">{{ .ID }}</td>
248 <td id="detail_name_{{ .ID }}">{{ .Name }}</td>
249 <td id="detail_description_{{ .ID }}">{{ .Description }}</td>
250 <td id="detail_weight_{{ .ID }}">{{ .Weight }}</td>
251 <td id="detail_image_{{ .ID }}">{{ .Image }}</td>
252 <td id="detail_supplier_{{ .ID }}">{{ .Supplier }}</td>
253 </tr>
254 {{ end }}
255 </tbody>
256 </table>
257
258 <div class="col">
259     <fieldset disabled>
260         <div class="mb-3">
261             <label for="detail_id" class="form-label">ID</label>
262             <input type="text" class="form-control" id="detail_id">
263         </div>
264     </fieldset>
265
266     <div class="mb-3">
267         <label for="detail_name" class="form-label">Имя</label>
268         <input type="text" class="form-control" id="detail_name">
269     </div>
270
271     <div class="mb-3">
272         <label for="detail_description" class="form-label">Описание</label>
273         <input type="text" class="form-control" id="detail_description">
274     </div>
275
276     <div class="mb-3">
277         <label for="detail_weight" class="form-label">Вес</label>
278         <input type="text" class="form-control" id="detail_weight">
279     </div>
280
281     <div class="mb-3">
282         <label for="detail_image" class="form-label">Превью</label>
283         <input type="text" class="form-control" id="detail_image">
284     </div>

```

```

285     <div class="mb-3">
286         <label for="detail_supplier" class="form-label">Поставщик</label>
287         <input type="text" class="form-control" id="detail_supplier">
288     </div>
289     <a class="btn btn-primary" role="button"
    ↳ onclick="updateNewDetail()">Применить</a>
290 </div>
291 </div>
292 </div>
293 {{ end }}
294
295 {{ if .Stocks }}
296     <div class="m-5">
297         <h1>Склады:</h1>
298         <div class="row align-items-start">
299
300             <table class="table table-striped col">
301                 <thead>
302                     <tr>
303                         <th scope="col">
304                             <a class="btn btn-primary btn-sm" role="button"
    ↳ onclick="newStock()"><i class="fas fa-plus"></i>
305                             </a>
306                         </th>
307                         <th scope="col">Номер</th>
308                         <th scope="col">Адрес</th>
309                         <th scope="col">Телефон</th>
310                     </tr>
311                 </thead>
312                 <tbody>
313                     {{ range .Stocks }}
314                         <tr>
315                             <td>
316                                 <div class="btn-group" role="group">
317                                     <a class="btn btn-warning btn-sm" role="button"
    ↳ onclick="editStock({{ .ID }})"><i
318                                         class="fas fa-edit"></i> </a>
319                                     <a class="btn btn-danger btn-sm" role="button"
    onclick="deleteRow({{ .ID }}, 'stocks')"><i class="fas
320                                         ↳ fa-trash"></i></a>
321                                 </div>
322                             </td>
323                             <td id="stock_id_{{ .ID }}">{{ .ID }}</td>
324                             <td id="stock_addr_{{ .ID }}">{{ .Address }}</td>
325                             <td id="stock_phone_{{ .ID }}">{{ .Phone }}</td>

```

```

326         </tr>
327     {{ end }}
328 </tbody>
329 </table>
330
331 <div class="col">
332     <fieldset disabled>
333         <div class="mb-3">
334             <label for="stock_id" class="form-label">ID</label>
335             <input type="text" class="form-control" id="stock_id">
336         </div>
337     </fieldset>
338
339     <div class="mb-3">
340         <label for="stock_addr" class="form-label">Адрес</label>
341         <input type="text" class="form-control" id="stock_addr">
342     </div>
343     <div class="mb-3">
344         <label for="stock_phone" class="form-label">Телефон</label>
345         <input type="text" class="form-control" id="stock_phone">
346     </div>
347     <a class="btn btn-primary" role="button"
348         ↪ onclick="updateNewStock()">Применить</a>
349 </div>
350 </div>
351 {{ end }}
352
353 {{ if .Suppliers }}
354     <div class="m-5">
355         <h1>Поставщики:</h1>
356         <div class="row align-items-start">
357
358             <table class="table table-striped col">
359                 <thead>
360                     <tr>
361                         <th scope="col">
362                             <a class="btn btn-primary btn-sm" role="button"
363                                 ↪ onclick="newSupplier()"><i
364                                     class="fas fa-plus"></i> </a>
365                         </th>
366                         <th scope="col">Номер</th>
367                         <th scope="col">Имя</th>
368                         <th scope="col">Адрес</th>
369                         <th scope="col">Телефон</th>

```

```

369     </tr>
370 </thead>
371 <tbody>
372 {{ range .Suppliers }}
373     <tr>
374         <td>
375             <div class="btn-group" role="group">
376                 <a class="btn btn-warning btn-sm" role="button"
377                     ↪ onclick="editSupplier({{ .ID }})"><i
378                         class="fas fa-edit"></i> </a>
379                 <a class="btn btn-danger btn-sm" role="button"
380                     onclick="deleteRow({{ .ID }}, 'suppliers')"><i
381                     ↪ class="fas fa-trash"></i></a>
382             </div>
383         </td>
384         <td id="supplier_id_{{ .ID }}">{{ .ID }}</td>
385         <td id="supplier_name_{{ .ID }}">{{ .Name }}</td>
386         <td id="supplier_addr_{{ .ID }}">{{ .Address }}</td>
387         <td id="supplier_phone_{{ .ID }}">{{ .Phone }}</td>
388     </tr>
389 {{ end }}
390 </tbody>
391 </table>
392
393 <div class="col">
394     <fieldset disabled>
395         <div class="mb-3">
396             <label for="supplier_id" class="form-label">ID</label>
397             <input type="text" class="form-control" id="supplier_id">
398         </div>
399     </fieldset>
400
401     <div class="mb-3">
402         <label for="supplier_name" class="form-label">Имя</label>
403         <input type="text" class="form-control" id="supplier_name">
404     </div>
405
406     <div class="mb-3">
407         <label for="supplier_addr" class="form-label">Адрес</label>
408         <input type="text" class="form-control" id="supplier_addr">
409     </div>
410
411     <div class="mb-3">
412         <label for="supplier_phone" class="form-label">Телефон</label>
413         <input type="text" class="form-control" id="supplier_phone">
414     </div>

```

```

411         <a class="btn btn-primary" role="button"
         ↳ onclick="updateNewSupplier()">Применить</a>
412     </div>
413 </div>
414 </div>
415 {{ end }}
416
417 {{ if .DetailStocks }}
418     <div class="m-5">
419         <h1>Детали на складах:</h1>
420         <div class="row align-items-start">
421
422             <table class="table table-striped col">
423                 <thead>
424                     <tr>
425                         <th scope="col">
426                             <a class="btn btn-primary btn-sm" role="button"
427                                 ↳ onclick="newDetailStock()"><i
428                                     class="fas fa-plus"></i> </a>
429
430                         <th scope="col">ID</th>
431                         <th scope="col">Деталь</th>
432                         <th scope="col">Склад</th>
433                         <th scope="col">Количество</th>
434                     </tr>
435                 </thead>
436                 <tbody>
437                     {{ range .DetailStocks }}
438                         <tr>
439                             <td>
440                                 <div class="btn-group" role="group">
441                                     <a class="btn btn-warning btn-sm" role="button"
442                                         ↳ onclick="editDetailStock({{ .ID }})"><i
443                                             class="fas fa-edit"></i> </a>
444                                     <a class="btn btn-danger btn-sm" role="button"1
445                                         onclick="deleteRow({{ .ID }}, 'detail_stocks')"><i
446                                             class="fas fa-trash"></i></a>
447                                 </div>
448                             </td>
449                             <td id="detail_stock_id_{{ .ID }}">{{ .ID }}</td>
450                             <td id="detail_stock_detail_id_{{ .ID }}">{{ .DetailID }}</td>
451                             <td id="detail_stock_stock_id_{{ .ID }}">{{ .StockID }}</td>
452                             <td id="detail_stock_count_{{ .ID }}">{{ .Count }}</td>
453                         </tr>
454                     {{ end }}

```

```

452         </tbody>
453     </table>
454
455     <div class="col">
456         <fieldset disabled>
457             <div class="mb-3">
458                 <label for="detail_stock_id" class="form-label">ID</label>
459                 <input type="text" class="form-control" id="detail_stock_id">
460             </div>
461         </fieldset>
462
463         <div class="mb-3">
464             <label for="detail_stock_detail_id" class="form-label">Деталь</label>
465             <input type="text" class="form-control" id="detail_stock_detail_id">
466         </div>
467         <div class="mb-3">
468             <label for="detail_stock_stock_id" class="form-label">Склад</label>
469             <input type="text" class="form-control" id="detail_stock_stock_id">
470         </div>
471         <div class="mb-3">
472             <label for="detail_stock_count" class="form-label">Кол-во</label>
473             <input type="text" class="form-control" id="detail_stock_count">
474         </div>
475         <a class="btn btn-primary" role="button"
476             ↪ onclick="updateNewDetailStock()">Применить</a>
477     </div>
478 </div>
479 {{ end }}
480
481 {{ if .ImagesIDs }}
482     <div class="m-5">
483         <h1>Загрузить изображение(не более 5 МБ)</h1>
484         <div class="input-group">
485             <input type="file" class="form-control" id="img" name="img" accept="image/*">
486             <a class="btn btn-primary" role="button" onclick="uploadImage()"><i class="fas
487                 ↪ fa-upload"></i> Загрузить</a>
488         </div>
489
490         <h1>Изображения</h1>
491         <table class="table">
492             <thead>
493                 <tr>
494                     <th scope="col">FileID</th>

```

```

495         <th scope="col"></th>
496     </tr>
497 </thead>
498 <tbody>
499     {{ range .ImagesIDs}}
500         <tr>
501             <td>{{ . }}</td>
502             <td></td>
504             <td><a class="btn btn-danger btn-sm" role="button"
505                 ↳ onclick="deleteImage('{{ . }}')"><i
506                     class="fas fa-trash"></i> </a></td>
507         </tr>
508     {{ end }}
509 </tbody>
510 </table>
511 </div>
512 {{ end }}
513 </body>
514 </html>

```

templates/error.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>403</title>
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link href="/static/css/bootstrap.min.css" rel="stylesheet">
8     <script src="/static/js/bootstrap.bundle.min.js"></script>
9     <style>
10         body {
11             background: #dedede;
12         }
13         .page-wrap {
14             min-height: 100vh;
15         }
16     </style>
17 </head>
18 <body>
19 <div class="page-wrap d-flex flex-row align-items-center">
20     <div class="container">
21         <div class="row justify-content-center">

```

```

22         <div class="col-md-12 text-center">
23             <span class="display-1 d-block">{{ .Code }}</span>
24             <div class="mb-4 lead">{{ .Description }}</div>
25             <a href="/" class="btn btn-link">Back to Home</a>
26         </div>
27     </div>
28 </div>
29 </div>
30 </body>
31 </html>

```

templates/stock.html

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Склад</title>
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link href="/static/css/bootstrap.min.css" rel="stylesheet">
8      <script src="/static/js/bootstrap.bundle.min.js"></script>
9      <style>
10         .limited-keepaspect {
11             max-width: 256px;
12             max-height: 256px;
13             width: auto;
14             height: auto;
15         }
16     </style>
17 </head>
18 <body>
19
20 <div class="m-5">
21     <div class="btn-group" role="group">
22         <a class="btn btn-primary" role="button" onclick="history.back()">Назад</a>
23         <a class="btn btn-primary" role="button" href="/">На главную</a>
24     </div>
25     <h1>Склад: {{ .Stock.Address }}</h1>
26     <table class="table">
27         <thead>
28             <tr>
29                 <th scope="col">Номер</th>
30                 <th scope="col">Телефон</th>
31             </tr>
32         </thead>
33         <tbody>

```



```

34     <tr>
35         <td>{{ .Stock.ID }}</td>
36         <td>{{ .Stock.Phone }}</td>
37     </tr>
38 </tbody>
39 </table>
40
41 {{ if .Details }}
42     <h1>Детали на этом складе:</h1>
43     <table class="table table-striped">
44         <thead>
45             <tr>
46                 <th scope="col">Номер</th>
47                 <th scope="col">Имя</th>
48                 <th scope="col">Описание</th>
49                 <th scope="col">Вес</th>
50                 <th scope="col">Количество</th>
51                 <th scope="col">Поставщик</th>
52                 <th scope="col">Превью</th>
53             </tr>
54         </thead>
55         <tbody>
56             {{ range .Details }}
57                 <tr>
58                     <td><a class="btn btn-link" href="/detail?id={{ .ID }}">{{ .ID
59                     ↳ }}</a></td>
60                     <td>{{ .Name }}</td>
61                     <td>{{ .Description }}</td>
62                     <td>{{ .Weight }}</td>
63                     <td>{{ .Count }}</td>
64                     <td><a class="btn btn-link" href="/supplier?id={{ .Supplier }}">{{
65                     ↳ .SupplierName }}</a></td>
66                     <td></td>
68                 </tr>
69             {{ end }}
70         </tbody>
71     </table>
72 {{ end }}
73
74 {{ if .Suppliers }}
75     {{ end }}
76 </div>

```

```
76 </body>
77 </html>
```

templates/detail.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Деталь</title>
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link href="/static/css/bootstrap.min.css" rel="stylesheet">
8     <script src="/static/js/bootstrap.bundle.min.js"></script>
9
10    <link href="/static/c3/c3.css" rel="stylesheet">
11    <script src="/static/c3/d3-5.8.2.min.js" charset="utf-8"></script>
12    <script src="/static/c3/c3.min.js"></script>
13
14    <style>
15        .limited-keepaspect {
16            max-width: 256px;
17            max-height: 256px;
18            width: auto;
19            height: auto;
20        }
21    </style>
22 </head>
23 <body>
24 <div class="m-5">
25     <div class="btn-group" role="group">
26         <a class="btn btn-primary" role="button" onclick="history.back()">Назад</a>
27         <a class="btn btn-primary" role="button" href="/">На главную</a>
28     </div>
29     <h1>Деталь: {{ .Detail.Name }}</h1>
30     <table class="table">
31         <thead>
32             <tr>
33                 <th scope="col">Номер</th>
34                 <th scope="col">Описание</th>
35                 <th scope="col">Вес</th>
36                 <th scope="col">Превью</th>
37             </tr>
38         </thead>
39         <tbody>
40             <tr>
41                 <td>{{ .Detail.ID }}</td>
```

```

42     <td>{{ .Detail.Description }}</td>
43     <td>{{ .Detail.Weight }}</td>
44     <td></td>
45 </tr>
46 </tbody>
47 </table>
48
49 {{ if .Supplier }}
50     <h1>Поставщик: {{ .Supplier.Name }}</h1>
51     <table class="table">
52         <thead>
53             <tr>
54                 <th scope="col">Номер</th>
55                 <th scope="col">Адрес</th>
56                 <th scope="col">Телефон</th>
57             </tr>
58         </thead>
59         <tbody>
60             <tr>
61                 <td><a class="btn btn-link" href="/supplier?id={{ .Supplier.ID }}">{{
    ↳ .Supplier.ID }}</a></td>
62                 <td>{{ .Supplier.Address }}</td>
63                 <td>{{ .Supplier.Phone }}</td>
64             </tr>
65         </tbody>
66     </table>
67 {{ end }}
68
69 {{ if .Stocks }}
70     <h1>Склады:</h1>
71     <div id="chart"></div>
72     <script>
73         const chart = c3.generate({
74             bindto: '#chart',
75             data: {
76                 columns: [
77                     {{ range .Stocks }}
78                     ['{{ .Address }}', {{ .Count }}],
79                     {{ end }}
80                 ],
81                 type: 'pie'
82             },
83             bar: {
84                 width: {

```

```

85         ratio: 0.5
86     }
87 }
88 });
89 </script>
90 <table class="table table-striped">
91     <thead>
92     <tr>
93         <th scope="col">Номер</th>
94         <th scope="col">Адрес</th>
95         <th scope="col">Телефон</th>
96         <th scope="col">Кол-во товара</th>
97     </tr>
98 </thead>
99 <tbody>
100     {{ range .Stocks }}
101     <tr>
102         <td><a class="btn btn-link" href="/stock?id={{ .ID }}">{{ .ID
103             ↳ }}</a></td>
104         <td>{{ .Address }}</td>
105         <td>{{ .Phone }}</td>
106         <td>{{ .Count }}</td>
107     </tr>
108     {{ end }}
109 </tbody>
110 </table>
111 {{ end }}
112 </div>
113 </body>
114 </html>

```

templates/master.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Details</title>
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link href="/static/css/bootstrap.min.css" rel="stylesheet">
8     <script src="/static/js/bootstrap.bundle.min.js"></script>
9 </head>
10 <body>
11 <div class="row align-items-start m-5">
12     <div class="col">

```

```

13     <h1>Детали:</h1>
14     <div class="list-group">
15         {{ range .Details }}
16             <a href="/detail?id={{ .ID }}" class="list-group-item
17                 ↳ list-group-item-action">{{ .Name }}</a>
18         {{ end }}
19     </div>
20
21 <div class="list-group col">
22     <h1>Поставщики:</h1>
23     <div class="list-group">
24         {{ range .Suppliers }}
25             <a href="/supplier?id={{ .ID }}" class="list-group-item
26                 ↳ list-group-item-action">{{ .Name }}</a>
27         {{ end }}
28     </div>
29 </div>
30 <div class="list-group col">
31     <h1>Склады:</h1>
32     <div class="list-group">
33         {{ range .Stocks }}
34             <a href="/stock?id={{ .ID }}" class="list-group-item
35                 ↳ list-group-item-action">{{ .Address }}</a>
36         {{ end }}
37     </div>
38 </div>
39 </div>
</body>
</html>

```

templates/supplier.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Поставщик</title>
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link href="/static/css/bootstrap.min.css" rel="stylesheet">
8     <script src="/static/js/bootstrap.bundle.min.js"></script>
9     <style>
10         .limited-keepaspect {
11             max-width: 256px;
12             max-height: 256px;
13             width: auto;

```

```

14         height: auto;
15     }
16 </style>
17 </head>
18 <body>
19
20 <div class="m-5">
21     <div class="btn-group" role="group">
22         <a class="btn btn-primary" role="button" onclick="history.back()">Назад</a>
23         <a class="btn btn-primary" role="button" href="/">На главную</a>
24     </div>
25     <h1>Поставщик: {{ .Supplier.Name }}</h1>
26     <table class="table">
27         <thead>
28             <tr>
29                 <th scope="col">Номер</th>
30                 <th scope="col">Адрес</th>
31                 <th scope="col">Телефон</th>
32             </tr>
33         </thead>
34         <tbody>
35             <tr>
36                 <td>{{ .Supplier.ID }}</td>
37                 <td>{{ .Supplier.Address }}</td>
38                 <td>{{ .Supplier.Phone }}</td>
39             </tr>
40         </tbody>
41     </table>
42
43     {{ if .Details }}
44     <h1>Детали, которые поставляет:</h1>
45     <table class="table table-striped">
46         <thead>
47             <tr>
48                 <th scope="col">Номер</th>
49                 <th scope="col">Имя</th>
50                 <th scope="col">Описание</th>
51                 <th scope="col">Вес</th>
52                 <th scope="col">Превью</th>
53             </tr>
54         </thead>
55         <tbody>
56             {{ range .Details }}
57                 <tr>

```

```

58         <td><a class="btn btn-link" href="/detail?id={{ .ID }}">{{ .ID
           ↳ }}</a></td>
59         <td>{{ .Name }}</td>
60         <td>{{ .Description }}</td>
61         <td>{{ .Weight }}</td>
62         <td></td>
63     </tr>
64     {{ end }}
65 </tbody>
66 </table>
67 {{ end }}
68
69 {{ if .Stocks }}
70     <h1>Склады, на которые поставляет:</h1>
71     <table class="table table-striped">
72         <thead>
73             <tr>
74                 <th scope="col">Номер</th>
75                 <th scope="col">Адрес</th>
76                 <th scope="col">Телефон</th>
77             </tr>
78         </thead>
79         <tbody>
80             {{ range .Stocks }}
81                 <tr>
82                     <td><a class="btn btn-link" href="/stock?id={{ .ID }}">{{ .ID
                       ↳ }}</a></td>
83                     <td>{{ .Address }}</td>
84                     <td>{{ .Phone }}</td>
85                 </tr>
86             {{ end }}
87         </tbody>
88     </table>
89     {{ end }}
90 </div>
91
92 </body>
93 </html>

```

models/Stock.go

```

1 package models
2
3 import "gorm.io/gorm"
4

```

```

5  type Stock struct {
6      gorm.Model
7      Address string `gorm:"address"`
8      Phone   string `gorm:"phone"`
9  }
10
11 type StockView struct {
12     Stock
13     Count uint `gorm:"count"`
14 }

```

models/Detail.go

```

1  package models
2
3  import "gorm.io/gorm"
4
5  type Detail struct {
6      gorm.Model
7      Name      string `gorm:"name"`
8      Description string `gorm:"description"`
9      Weight     float64 `gorm:"weight"`
10     Image      string `gorm:"image"`
11     Supplier   uint   `gorm:"supplier_id"`
12 }
13
14 type DetailView struct {
15     Detail
16     Count      uint   `gorm:"count"`
17     SupplierName string `gorm:"supplier_name"`
18 }

```

models/Supplier.go

```

1  package models
2
3  import "gorm.io/gorm"
4
5  type Supplier struct {
6      gorm.Model
7      Name      string `gorm:"name"`
8      Address    string `gorm:"address"`
9      Phone      string `gorm:"phone"`
10 }

```

models/DetailStock.go

```

1  package models
2

```



```

3 import "gorm.io/gorm"
4
5 type DetailStock struct {
6     gorm.Model
7     DetailID uint `gorm:"detail_id"`
8     StockID   uint `gorm:"supplier_id"`
9     Count     uint `gorm:"count"`
10 }

```

3 Экранные формы

The screenshot shows a web browser window with the address bar displaying 'git.tujh.xyz'. The main content area contains three tables arranged horizontally:

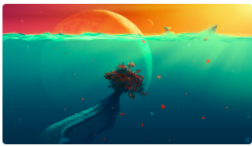
Деталь1
Деталь2
Деталь3
Деталь4
Деталь5

Поставщик1
Поставщик2

Склад1
Склад2
Склад3
Склад4

[Назад](#)
[На главную](#)

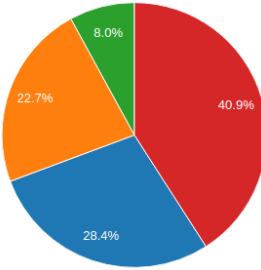
Деталь: Деталь1

Номер	Описание	Вес	Превью
1	Описание Детали1	12	

Поставщик: Поставщик1

Номер	Адрес	Телефон
1	Адрес Поставщика1	+79998887766

Склады:






Номер	Адрес	Телефон	Кол-во товара
1	Склад1	+7987654321	250

[Назад](#)
[На главную](#)

Склад: Склад1

Номер	Телефон
1	+7987654321

Детали на этом складе:





Номер	Имя	Описание	Вес	Количество	Поставщик	Превью
1	Деталь1	Описание Детали1	12	250	Поставщик1	
3	Деталь3	Описание Детали3	15.36	50	Поставщик2	
4	Деталь4	Описание Детали4	30.08	80	Поставщик2	

НазадНа главную

Поставщик: Поставщик1

Номер	Адрес	Телефоон
1	Адрес Поставщика1	+79998887766

Детали, которые поставляет:

Номер	Имя	Описание	Вес	Превью
1	Деталь1	Описание Детали1	12	
2	Деталь2	Описание Детали2	30	
3	Деталь3	Описание Детали3	15.36	
4	Деталь4	Описание Детали4	30.08	

Admin Главная Детали Поставщики Склады Детали на складах Изображения

Детали:

Номер

Имя

Описание

Вес

Превью

Поставщик

<div><div></div><div></div></div>	1	Деталь1	Описание Детали1	12	CzFlyGVOTdMNgoiPzk56lkP6t_g=	1
<div><div></div><div></div></div>	2	Деталь2	Описание Детали2	30	37bDNU2vUdY1cfcWEBdhtx_sRLc=	1
<div><div></div><div></div></div>	3	Деталь3	Описание Детали3	15.36	gV1YrIfF7CRNKejgX_CrtME6r3k=	2
<div><div></div><div></div></div>	4	Деталь4	Описание Детали4	30.08	mgrz1V2NKYkwBOLBBRfpAGgVYsg=	2
<div><div></div><div></div></div>	5	Деталь5	Описание Детали5	120	eEX25-eTXAHi9o9lhEdqThyNxw8=	2

ID

Имя

Описание

Вес

Превью

Поставщик

Применить

git.tujh.xyz/admin?token= &table=suppliers&offset=

Admin Главная Детали Поставщики Склады Детали на складах Изображения

Поставщики:

+

✎

🗑

Номер	Имя	Адрес	Телефон
1	Поставщик1	Адрес Поставщика1	+79998887766
2	Поставщик2	Адрес Поставщика2	+77776665544

ID

Имя

Адрес

Телефон

Применить

git.tujh.xyz/admin?token= &table=stocks&offset=

Admin Главная Детали Поставщики Склады Детали на складах Изображения

Склады:

+

✎

🗑

Номер	Адрес	Телефон
1	Склад1	+7987654321
2	Склад2	+71234567890
3	Склад3	+78889996543
4	Склад4	+74567892134

ID

Адрес

Телефон

Применить

git.tujh.xyz/admin?token=qaedwscxz&table=detail_stocks&offset=

Admin Главная Детали Поставщики Склады Детали на складах Изображения

Детали на складах:

+

✎

🗑

ID	Деталь	Склад	Количество
1	1	1	250
2	1	2	200
3	2	3	20
4	3	4	20
5	2	4	50
6	4	2	120
7	3	1	50
8	4	1	80
9	1	3	70
10	1	4	360

ID

Деталь

Склад

Кол-во

Применить

Admin Главная Детали Поставщики Склады Детали на складах Изображения

Загрузить изображение(не более 5 МБ)

Choose File

No file chosen

Загрузить

Изображения

FileID	Превью	
37bDNU2vUdY1cfcWEBdhtx_sRLc=		
CzFlyGVOTdMNgoiPzk56lkP6t_g=		
eEX25-eTXAHi9o9lhEdqThyNxx8=		
gV1Yr1fF7CRNKejgX_CrtME6r3k=		
mqrz1V2NKYkwBOLBBRfpAGqVYsg=		