



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

### **ПРАКТИЧЕСКАЯ РАБОТА №3**

по дисциплине «Проектирование и разработка серверных частей интернет-  
ресурсов»

**Студент группы ИКБО-21-23**

**Зеленков Н.А.**

---

(подпись студента)

**Руководитель практической работы**

**Благирев М.М.**

---

(подпись руководителя)

Работа представлена

«\_\_\_» \_\_\_\_\_ 2025 г.

Допущен к работе

«\_\_\_» \_\_\_\_\_ 2025 г.

Москва 2025

## **ЦЕЛЬ РАБОТЫ**

В задании предлагается создать сложную серверную конфигурацию, состоящую из связки apache+nginx+php+База данных. Возможно использование связки apache+php как единый компонент. В данной конфигурации предполагается создание как минимум 3 элементов(контейнеров) или использование как основы серверной конфигурации, созданной в практической работе №1. В этой конфигурации предполагается акселерированное проксирование без кэширования. Схематично предполагаемый алгоритм работы изображен на рисунке.

Предполагается, что сервер nginx будет отображать статический контент, а apache динамический и в связке мы получим быстродейственную и эффективную систему.

Также по необъяснимым обстоятельствам следует изменить root директории с базовых на другие. Для доступа к администрированию предусмотреть базовую авторизацию и аутентификацию с применением htaccess и хранением пользователей в БД (без использования htpasswd).

Для тестирования данной конфигурации предполагается создать тестовое веб-приложение на тему по варианту:

1. Кофейня
2. Строительный магазин
3. Автосервис
4. Магазин электроники
5. Портфолио
6. Библиотека
7. Ресторан
8. Погода
9. Ломбард
- 10.Магазин игрушек
- 11.Индивидуальная тема

Тестовое веб-приложение предполагает создание как минимум 2 веб-страниц со статическим контентом и двух веб-страниц с динамическим контентом: взятым из базы данных, например.

# ХОД РАБОТЫ

Для выполнения программы были разработаны несколько файлов (см. рис. 1).

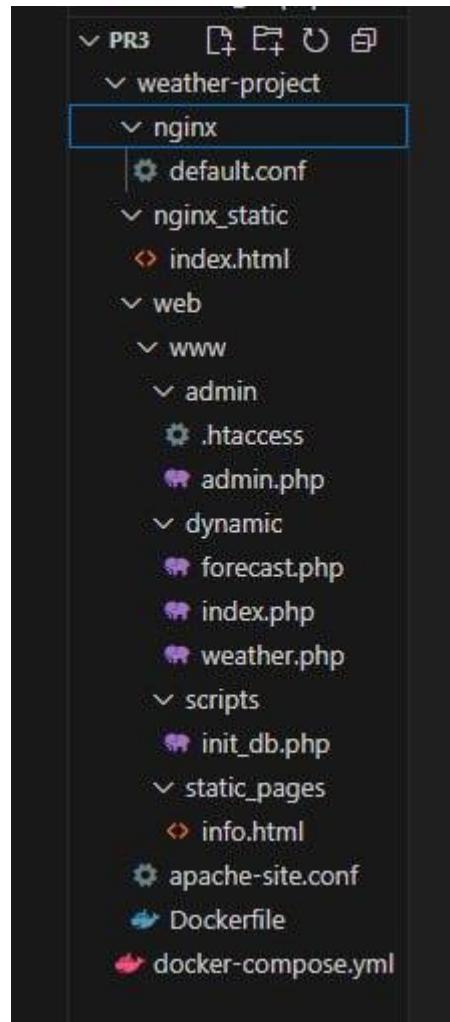


Рисунок 1 – Структура проекта

Были разработаны четыре статические страницы (см. рис. 2-3).

```
1 <!doctype html>
2 <html>
3 <head><meta charset="utf-8"><title>Weather – статический фронт</title>
4 <link rel="stylesheet" href="/static/assets/style.css">
5 </head>
6 <body>
7 <h1>Welcome – Static Weather Landing</h1>
8 <p>Это статическая главная страница, обслуживаемая nginx.</p>
9 <p><a href="/dynamic/index.php">Перейти к динамике (apache+php)</a></p>
10 </body></html>
11
```

Рисунок 2 – Статическая страница index.html

```
web > www > static_pages > info.html > ...
2   <html lang="ru">
3     <head>
7       <style>
23       a {
26         }
27     <a: hover {
28       text-decoration: underline;
29     }
30   </style>
31 </head>
32 <body>
33   <header>
34     <h1>0 нашем погодном приложении</h1>
35   </header>
36
37   <main>
38     <p>
39       Это тестовое веб-приложение создано для демонстрации работы связки
40       <strong>nginx + apache + php + MySQL</strong>.
41     </p>
42     <p>
43       <strong>nginx</strong> обслуживает статический контент (страницы вроде этой),
44       а <strong>Apache + PHP</strong> – динамический (данные из базы, прогнозы и т.д
45     </p>
46     <p>
47       В качестве примера динамического контента вы можете перейти на:
48       <a href="/dynamic/index.php">страницу списка городов</a>.
49     </p>
50   </main>
51
52   <footer>
53     &copy; 2025 WeatherApp – демо-проект для учебных целей.
54   </footer>
55 </body>
56 </html>
57
```

Рисунок 3 – Статическая страница info.html

Для динамической работы написаны файлы (см. рис. 4–6).

```
web > www > dynamic > index.php
1   <?php
2   $dbh = new mysqli(getenv('DB_HOST'), getenv('DB_USER'), getenv('DB_PASS'), getenv('DB_NAME'));
3   $res = $dbh->query("SELECT id, city, temp_c, summary FROM cities ORDER BY city");
4   ?>
5   <!doctype html>
6   <html><head><meta charset="utf-8"><title>Список городов – Погода</title></head>
7   <body>
8     <h1>Города</h1>
9     <ul>
10    <?php while($r = $res->fetch_assoc()): ?>
11    <li><a href="/dynamic/weather.php?id=<?= $r['id'] ?>"><?= htmlspecialchars($r['city']) ?>
12      <?= $r['temp_c'] ?>°C – <?= htmlspecialchars($r['summary']) ?></a></li>
13    <?php endwhile; ?>
14  </ul>
15 </body></html>
16
```

Рисунок 4 – динамическая страница index.php

```

web > www > dynamic > weather.php
1  <?php
2  $dbh = new mysqli(getenv('DB_HOST'), getenv('DB_USER'), getenv('DB_PASS'), getenv('DB_NAME'));
3  if ($dbh->connect_error) {
4      die("DB connection error");
5  }
6  $id = isset($_GET['id']) ? (int)$_GET['id'] : 0;
7  $stmt = $dbh->prepare('SELECT city, temp_c, summary, updated_at FROM cities WHERE id = ?');
8  $stmt->bind_param('i', $id);
9  $stmt->execute();
10 $stmt->bind_result($city, $temp_c, $summary, $updated_at);
11 if ($stmt->fetch()) {
12     // ok
13 } else {
14     header("HTTP/1.0 404 Not Found");
15     echo "City not found";
16     exit;
17 }
18 $stmt->close();
19 ?>
20 <!doctype html>
21 <html><head><meta charset="utf-8"><title><?=htmlspecialchars($city)?></title></head>
22 <body>
23 <h1>Погода: <?=htmlspecialchars($city)?></h1>
24 <p>Температура: <?=htmlspecialchars($temp_c)?> °C</p>
25 <p>Сводка: <?=htmlspecialchars($summary)?></p>
26 <p>Обновлено: <?=htmlspecialchars($updated_at)?></p>
27 <p><a href="/dynamic/index.php">Назад к списку</a></p>
28 </body></html>
29

```

Рисунок 5 – динамическая страница weather.php

```

web > www > dynamic > forecast.php
1  <?php
2  // Здесь можно выводить прогноз из другой таблицы – для примера статический текст
3  ?>
4  <!doctype html><html><head><meta charset="utf-8"><title>Прогноз</title></head>
5  <body>
6  <h1>Простая страница прогноза</h1>
7  <p>Прогноз на завтра: преимущественно ясно. (Данные для демонстрации)</p>
8  </body></html>
9

```

Рисунок 6 – динамическая страница forecast.php

Для настройки прокси на базе nginx используется файл default.conf (см. рис. 7).

```
nginx > default.conf
  1 ✓ server {
  2     listen 80;
  3     server_name _;
  4
  5     root /srv/www_static;
  6     index index.html;
  7
  8     access_log /var/log/nginx/access.log;
  9     error_log /var/log/nginx/error.log;
 10
 11    location ~* \.(?:css|js|png|jpg|jpeg|gif|ico|svg|woff2?)$ {
 12        try_files $uri =404;
 13        access_log off;
 14        expires 7d;
 15    }
 16
 17    location /static/ {
 18        alias /srv/www_static/;
 19        try_files $uri $uri/ =404;
 20        access_log off;
 21        expires 7d;
 22    }
 23
 24    location / {
 25        try_files $uri $uri/ @backend;
 26    }
 27
 28    location @backend [
 29        proxy_pass http://web:80;
 30        proxy_set_header Host $host;
 31        proxy_set_header X-Real-IP $remote_addr;
 32        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 33        proxy_set_header X-Forwarded-Proto $scheme;
 34
 35        proxy_buffering off;
 36        proxy_cache_bypass $http_pragma $http_authorization;
 37        proxy_ignore_client_abort on;
 38        proxy_read_timeout 90;
 39    ]
 40 }
 41 }
```

Рисунок 7 – default.conf

Для сборки проекта используется докер (см. рис. 8).

```
❸ docker-compose.yml
1  version: '3.8'
2  services:
3    db:
4      image: mysql:8.0
5      restart: unless-stopped
6      environment:
7        MYSQL_ROOT_PASSWORD: rootpass
8        MYSQL_DATABASE: weatherdb
9        MYSQL_USER: appuser
10       MYSQL_PASSWORD: apppass
11      volumes:
12        - db_data:/var/lib/mysql
13      networks:
14        - webnet
15
16    web:
17      build: ./web
18      depends_on:
19        - db
20      environment:
21        DB_HOST: db
22        DB_NAME: weatherdb
23        DB_USER: appuser
24        DB_PASS: apppass
25      volumes:
26        - ./web/www:/var/www/weather_site    # код приложения
27      networks:
28        - webnet
29      expose:
30        - "80"
31
32    nginx:
33      image: nginx:stable
34      depends_on:
35        - web
36      ports:
37        - "80:80"
38      volumes:
39        - ./nginx/default.conf:/etc/nginx/conf.d/default.conf:ro
40        - ./nginx_static:/srv/www_static:ro          # нестандартная root для nginx
41      networks:
42        - webnet
```

Рисунок 8 – Docker-compose.yml

## ТЕСТИРОВАНИЕ

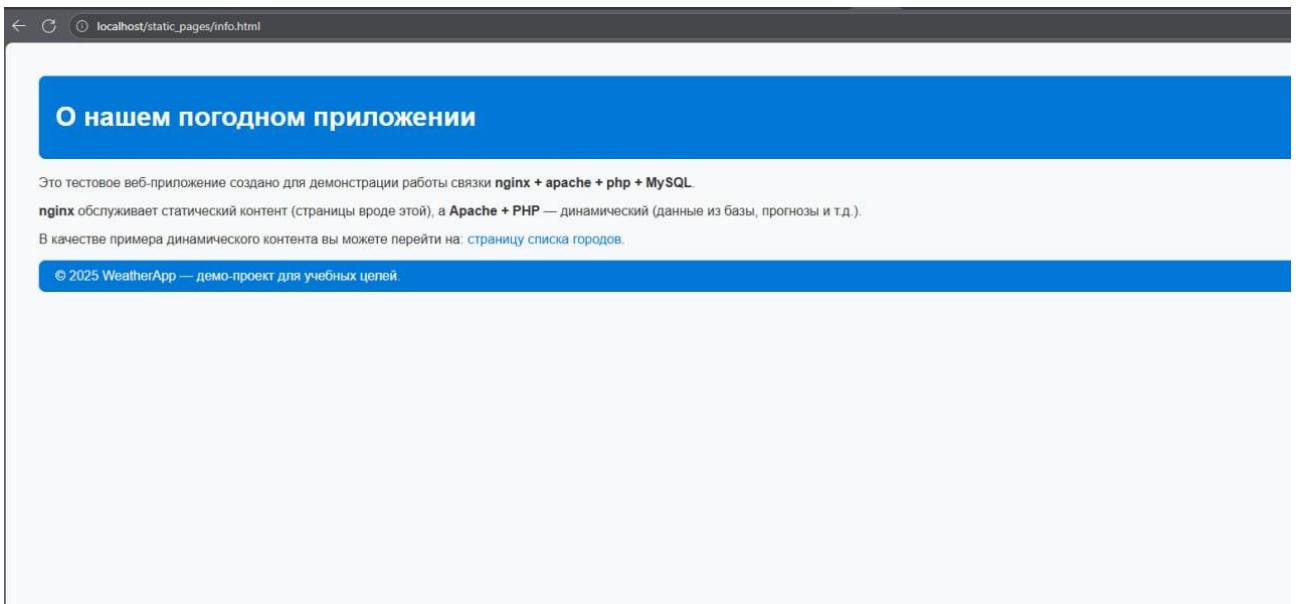


Рисунок 9 – статическая инфо страница

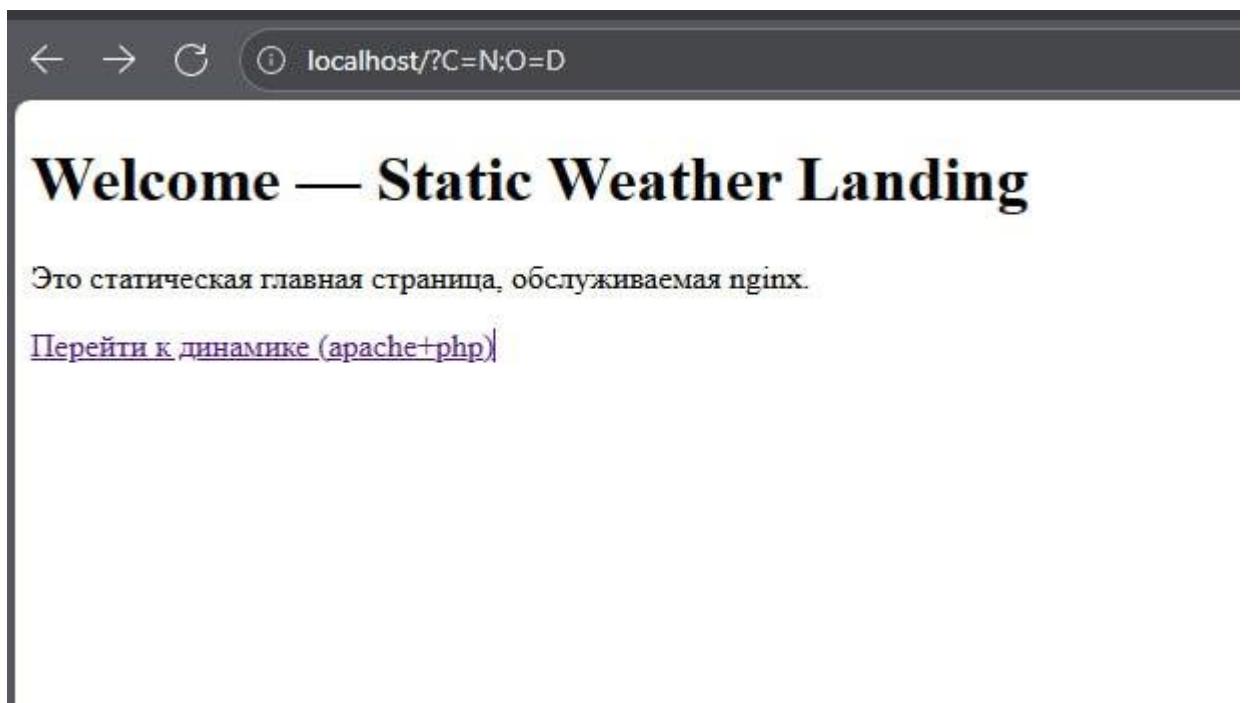


Рисунок 10 – статическая страница под nginx



Рисунок 11 – динамическая страница с инфой из бд



Рисунок 12 – динамическая страница

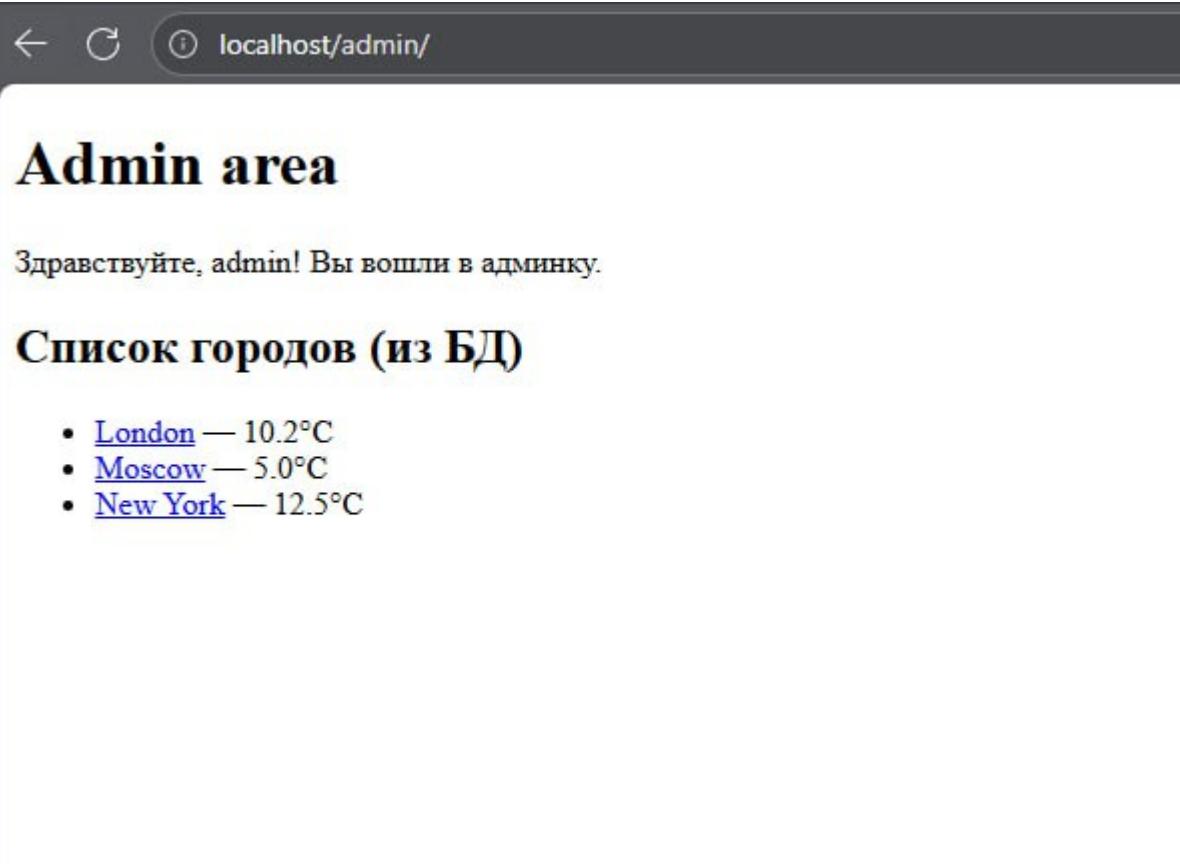


Рисунок 13 – админская страница

## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Документация // PHP URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 10.09.2025). – Текст: электронный.
2. Статья о назначении докера простыми <https://habr.com/ru/post/309556/> (дата обращения: 10.09.2025).
3. словами: Методические указания по выполнению практической работы: <https://online-edu.mirea.ru/mod/resource/view.php?id=508421> (дата обращения: 10.09.2025).
4. Официальная документация докера: <https://docs.docker.com/> (дата обращения: 10.09.2025).
5. Более сложная и подробная статья <https://habr.com/ru/post/277699/> (дата обращения: 10.09.2025).