

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 3

з дисципліни “Основи web-програмування”

тема “**Багатосторінковий веб-сайт**”

|  |  |  |
| --- | --- | --- |
| Виконав  студент II курсу  групи КП-92  Мельник Ігор Сергійович  (*прізвище, ім’я, по батькові*)  варіант №8 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

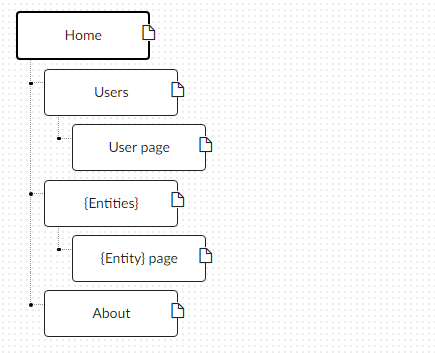
**Мета роботи**

Навчитись створювати веб-сайт, що надає CRUD доступ до ресурсів веб-сервера.  
Ознайомитись із шаблонізаторами веб-сторінок для генерації контенту.  
Навчитись створювати HTML-форми для взаємодії користувача із веб-сайтом та обробляти на сервері внесені користувачем дані.

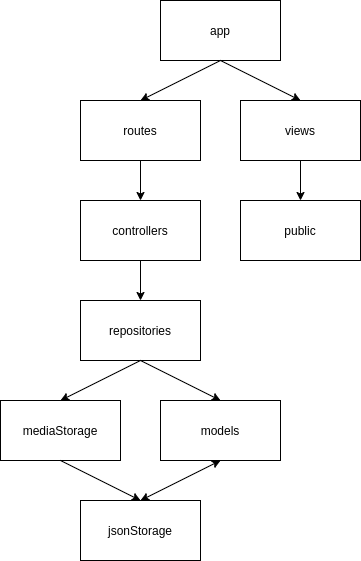
**Постановка завдання**

Реалізувати багатосторінковий веб-сайт із заданою структурою розділів:

Забезпечити можливість пошуку, пагінації, додавання та видалення сутностей за варіантом.



**Діаграма залежності модулів**

****

**Код всіх всіх нових та змінених модулів**

|  |
| --- |
| **app.js** |
| const express = require('express');  const bodyParser = require('body-parser');  const mustache = require('mustache-express')  const path = require('path');  const morgan = require('morgan');  const busboyBodyParser = require('busboy-body-parser');  const userRouter = require('./routes/users');  const operatorRouter = require('./routes/operators');  const mediaRouter = require('./routes/media.js');  const app = express();  const viewsDir = path.join(\_\_dirname, 'views');  const partialsDir= path.join(viewsDir,'partials');  app.engine('mst', mustache(partialsDir));  app.set('views', viewsDir);  app.set('view engine', 'mst');  app.use(morgan('dev'));  app.use(busboyBodyParser({  limit: '10mb',  multi: true,  }));  app.use(express.static('public'));  app.use(bodyParser.urlencoded({ extended: true }));  app.get('/', function(req, res) {  res.render('index');  });  app.get('/about', function(req, res) {  res.render('about');  });  app.get('/data/media/:url', function(req, res) {  res.sendFile(`${\_\_dirname}/data/media/${req.params.url}`);  });  app.use('/users', userRouter);  app.use('/operators', operatorRouter);  app.use('/api/media', mediaRouter);  app.listen(3005, function () { console.log('Server is ready'); }); |

|  |
| --- |
| **routes/operators.js** |
| **const express = require('express');**  const router = express.Router();  const operatorController = require('../controllers/operators.js');  const bodyParser = require('body-parser');  const jsonParser = bodyParser.json();  /\*\*  \* @route GET /api/operators  \* @group Operators - operator operations  \* @param {integer} page.query - page number  \* @param {integer} per\_page.query - items per page  \* @returns {Array.<Tour\_operator>} 200 - a page with operators  \*/  /\*\*  \* @route GET /api/operators/{id}  \* @group Operators - operator operations  \* @param {integer} id.path.required - id of operator  \* @returns {Tour\_operator.model} 200 - Operator object  \* @returns {Error} 404 - Operator not found  \*/  /\*\*  \* @route POST /api/operators  \* @group Operators - operator operations  \* @param {Tour\_operator.model} id.body.required - new Operator Object  \* @returns {Tour\_operator.model} 201 - Created operator object  \* @returns {Error} 400 - Invalid field data  \*/  /\*\*  \* @route PUT /api/operators  \* @group Operators - operator operations  \* @param {Tour\_operator.model} object.body.required - new Operator Object  \* @returns {Tour\_operator.model} 200 - Updated game object  \* @returns {Error} 400 - Invalid field data  \* @returns {Error} 404 - Operator not found  \*/  /\*\*  \* @route DELETE /api/operators/{id}  \* @group Operators - operator operations  \* @param {integer} id.path.required - id of the Operator  \* @returns {Tour\_operator.model} 200 - deleted Operator object  \* @returns {Error} 404 - Operator not found  \*/  router  .get("/new", function (req, res) {  res.render('new');  })  .get("/:id", operatorController.getTour\_operatorById)  .get("", operatorController.getTour\_operators)  .post("/", jsonParser, operatorController.addTour\_operator)  .post("/:id", operatorController.deleteTour\_operator);  module.exports = router; |

|  |
| --- |
| **controllers/users.js** |
| const user\_rep = require('../repositories/userRepository.js');  const { use } = require('../routes/users.js');  const userRepository = new user\_rep("data/users.json");  module.exports = {  getUsers(req, res) {  const users = userRepository.getUsers();  res.render('users', {users});  },  getUserById(req, res) {  const userID = Number(req.params.id);  const user = userRepository.getUserById(userID);  res.render('user',user);  }  }; |

|  |
| --- |
| **controllers/operators.js** |
| const op\_rep = require('../repositories/operatorRepository.js');  const operatorRepository = new op\_rep("data/tour\_operators.json");  const moment = require('moment');  const media\_rep = require('../repositories/mediaRepository.js');  const mediaRepository = new media\_rep('data/media/', 'data/media.json');  const stringValidator = value => typeof (value) === "string";  const integerValidator = value => {  if (Number.isInteger(Number(value)) && Number(value) >= 0) {  return true;  }  return false;  };  const dateValidator = value => moment(value, moment.ISO\_8601).isValid();  const validators = {  id: integerValidator,  name: stringValidator,  countries: stringValidator,  amount\_tours: integerValidator,  amount\_departments: integerValidator,  foundation\_date: dateValidator,  url: stringValidator  };  const errorMessages = {  id: "id should be inputed as integer",  name: "Name field should be string",  countries: "Countries field should be string",  amount\_tours: "Amount of yours should be inputed as integer",  amount\_departments: "Amount of departments should be integer",  foundation\_date: "Please enter date in ISO 8601 format ",  url: "Should be a string"  };  module.exports = {  getTour\_operators(req, res) {  let operators = operatorRepository.getTour\_operators();  let input = req.query.name;  if (input != undefined && input != "") {  for (let i = 0; i < operators.length; i++) {  if (!operators[i].name.includes(input)) {  operators.splice(i, 1);  i--  }  }  input = `Результат пошуку операторів за назвою "${input}"`;  }  else {  input = "Результат пошуку усіх операторів";  }  const page\_size = 2;  let page\_number = Number.parseInt(req.query.page);  if (req.query.page === undefined) {  page\_number = 1;  }  const items\_total = operators.length;  if (!Number.isInteger(page\_number) || page\_number < 1 || page\_size \* (page\_number - 1) >= items\_total) {  res.render('operators', { error:"page not found",page: `${page\_number}/${Math.ceil(items\_total / page\_size)}`, prev: page\_number - 1, next: page\_number + 1, input,name:req.query.name });  }  else {  operators = operators.slice(page\_size \* (page\_number - 1), page\_size \* (page\_number));  res.render('operators', { operators, page:`${page\_number}/${Math.ceil(items\_total / page\_size)}`, prev: page\_number - 1, next: page\_number + 1, input ,name:req.query.name});  }  },  getTour\_operatorById(req, res) {  const operatorID = Number(req.params.id);  if (!Number.isInteger(operatorID) || operatorID < 1) {  return res.status(400).json({ Error: "Operator id should be natural number" });  }  try {  const op = operatorRepository.getTour\_operatorById(operatorID);  res.render('operator', op);  }  catch (error) {  return res.status(404).json({ Error: error.message });  }  },  addTour\_operator(req, res) {  res.set("Content-type", "application/json");  const operator = req.body;  const properties = ["id", "name", "countries", "amount\_tours", "amount\_departments", "foundation\_date", "url"];  if (!operator.name || !operator.countries) return res.status(400).json({ Error: "Operator must contain 'name' and 'countries' properties" });  for (const [key, value] of Object.entries(operator)) {  if (!properties.includes(key)) return res.status(400).json({ Error: `property "${key}" doesnt exist on operator entity` });  if (!validators[key](String(value))) return res.status(400).json({ message: errorMessages[key] });  }  mediaRepository.addMedia(req.files.photoFile[0]);  operator.url=`/data/media/${req.files.photoFile[0].name}`;  const id = operatorRepository.addTour\_operator(operator);  res.redirect(`/operators/${id}`)  },  updateTour\_operator(req, res) {  res.set("Content-type", "application/json");  const operator = req.body;  const properties = ["id", "name", "countries", "amount\_tours", "amount\_departments", "foundation\_date"];  if (!operator.name || !operator.countries) return res.status(400).json({ Error: "Operator must contain 'name' and 'countries' properties" });  for (const [key, value] of Object.entries(operator)) {  if (!properties.includes(key)) return res.status(400).json({ Error: `property "${key}" doesnt exist on operator entity` });  if (!validators[key](String(value))) return res.status(400).json({ message: errorMessages[key] });  }  operatorRepository.updateTour\_operator(operator);  try {  const new\_op = operatorRepository.getTour\_operatorById(operator.id);  return res.status(200).json(new\_op);  }  catch (error) {  return res.status(404).json({ Error: error.message });  }  },  deleteTour\_operator(req, res) {  res.set("Content-type", "application/json");  let id = parseInt(req.params.id);  try {  const op = operatorRepository.getTour\_operatorById(id);  operatorRepository.deleteTour\_operator(op.id);  res.redirect('/operators');  }  catch (error) {  return res.status(404).json({ Error: error.message });  }  }  }; |

|  |
| --- |
| **views/index.mst** |
| <!DOCTYPE html>  <html>  <head>  <title>Web</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <p style="font-size:25pt">  Web - тестовий сайт з можливістю роботи із користувачами та операторами</p>  <img src="/images/web.png" alt="Web image width="200" height="150">  </div>  {{> footer}}  </body>  </html> |

|  |
| --- |
| **views/users.mst** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Users</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <table class=" styled-table">  <caption font-size=40pt> Таблиця користувачів</caption>  <thead>  <tr>  <th>Login</th>  <th>Fullname</th>  <th>Redisteged</th>  </tr>  </thead>  <tbody>  {{#users}}  <tr>  <td><a href='/users/{{id}}'>{{login}}</a></td>  <td>{{fullname}}</td>  <td>{{registeredAt}}</td>  </tr>  {{/users}}  </tbody>  </table>  </div>  {{> footer}}  </body>  </html> |

|  |
| --- |
| **views/user.mst** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Web</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <img src={{avaUrl}} alt="User image" width="200" height="150">  <p> {{fullname}}</p>  <p> ({{login}}) {{registeredAt}}</p>  <p> {{bio}}</p>  <div>  {{> footer}}  </body>  </html> |

|  |
| --- |
| **views/operators.mst** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Users</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <h1>Список туроператорів</h1>  <ul class="circle">  {{#operators}}  <li><a href='/operators/{{id}}'>{{name}}</a></li>  {{/operators}}  {{^operators}}  <li>{{error}}</li>  {{/operators}}  </ul>  <p>{{input}}</p>  <form style="float:right;position: relative;left:100pt;top=0">  Пошук за назвою:<br>  <input type="text" name="name"><br>  <input type="submit" value="Submit">  </form>  <a class="post" href="/operators/new">ADD</a>  </div>  <div class="pagination">  <a class="prev" href="/operators?page={{prev}}&name={{name}}">prev</a>  <a class="current">{{page}}</a>  <a class="next" href="/operators?page={{next}}&name={{name}}">next</a>  </div>  {{> footer}}  </body>  </html> |

|  |
| --- |
| **views/operator.mst** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Web</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <table class=" styled-table">  <caption font-size=40pt> Оператор</caption>  <thead>  <tr>  <th>Name</th>  <th>Countries</th>  <th>Amount of tours</th>  <th>Amount of departments</th>  <th>Foundation date</th>  </tr>  </thead>  <tbody>  <tr>  <td>{{name}}</td>  <td>{{countries}}</td>  <td>{{amount\_tours}}</td>  <td>{{amount\_departments}}</td>  <td>{{foundation\_date}}</td>  </tr>  </tbody>  </table>  <div style="float: right;  position: relative;">  <img src={{url}} alt="Operator image" width="200" height="150">  <a href ={{url}}>Photo</a>  </div>  <form method="POST" action="/operators/{{id}}">  <input type="submit" value="delete">  </form>  <div>  {{> footer}}  </body>  </html> |

|  |
| --- |
| **views/new.mst** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Users</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <form method="POST" action="/operators/" enctype="multipart/form-data">  Name:<br>  <input type="text" name="name" ><br>  Countries:<br>  <input type="text" name="countries" ><br>  Amount of tours:<br>  <input type="number" name="amount\_tours" ><br>  Amount of departments:<br>  <input type="number" name="amount\_departments" ><br>  Foundation date:<br>  <input type="text" name="foundation\_date" ><br>  <input type="file" name="photoFile"><br>  <input type="submit" value="Submit">  </form>  </div>  {{> footer}}  </body>  </html> |

|  |
| --- |
| **views/about.mst** |
| <!DOCTYPE html>  <html>  <head>  <title>About</title>  {{> head}}  </head>  <body>  {{> header}}  <div class=content>  <h1> Web - тестовий сайт з можливістю роботи із користувачами та операторами</h1>  <h2>Автор</h2>  <p>Мельник Ігор Сергійович</p>  <p>telegram: <a href="https://t.me/typical\_sirius">@typical\_sirius</a></p>  </div>  {{> footer}}  </body>  </html> |

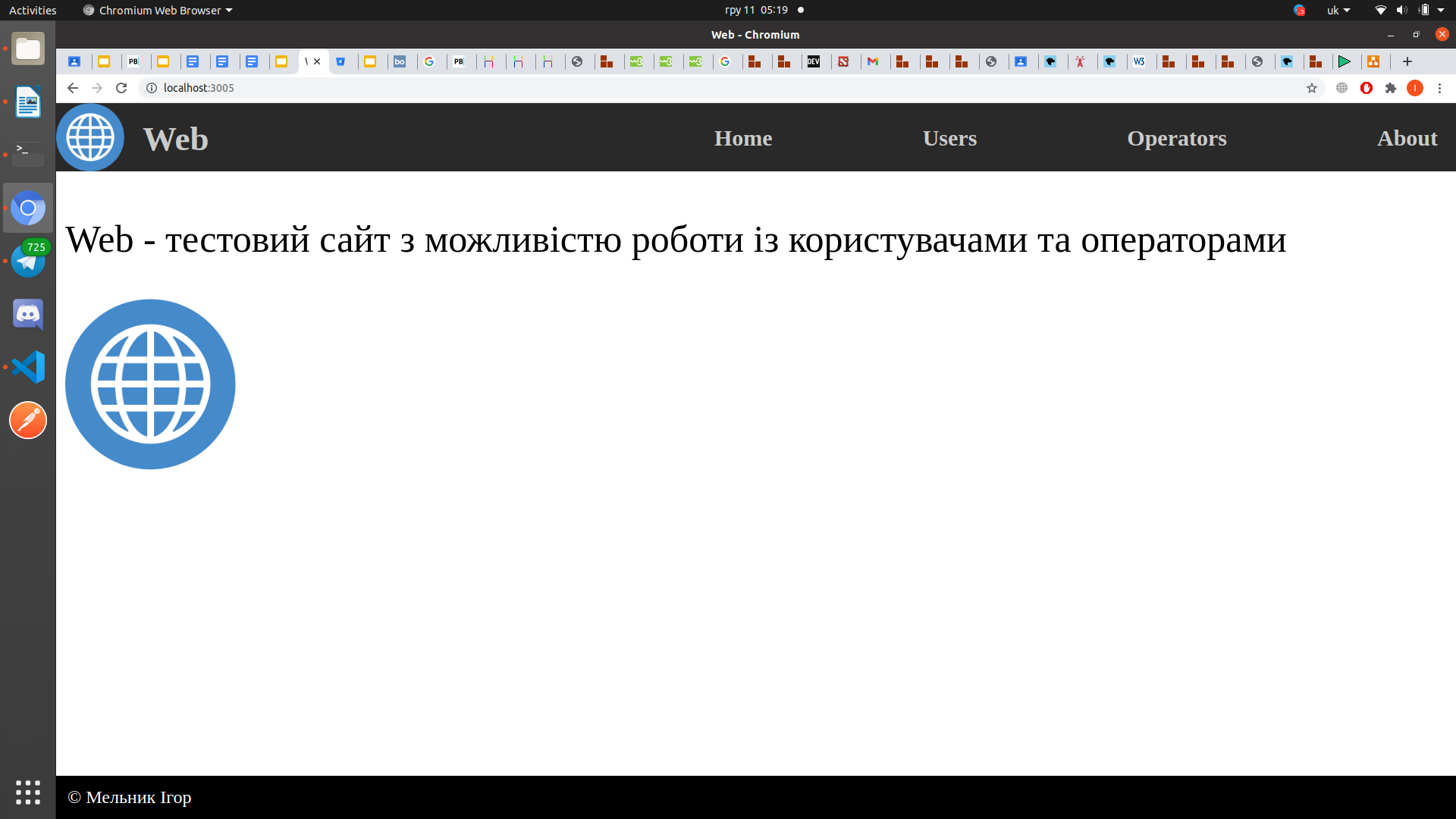
|  |
| --- |
| **views/partials/head.mst** |
| <meta charset="utf-8">  <link type="text/css" href="/stylesheets/style.css" rel="stylesheet" /> |

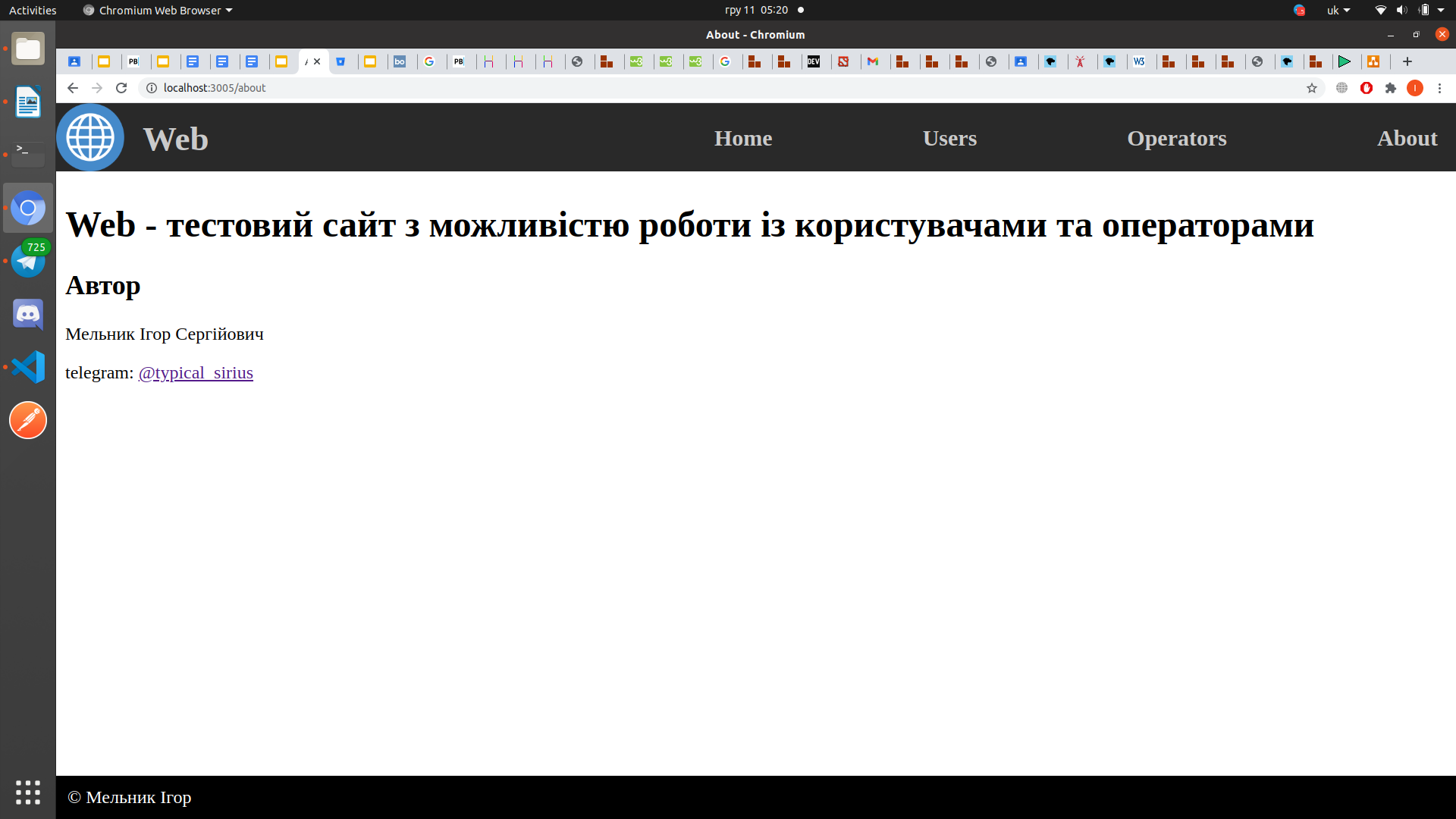
|  |
| --- |
| **views/partials/header.mst** |
| <ul class="header">  <li class="logo">  <img src="/images/web.png">  <a href="/">Web</a> </li>  <li class="menu">  <a href="/">Home</a>  <a href="/users">Users</a>  <a href="/operators">Operators</a>  <a href="/about">About</a></li>  </ul> |

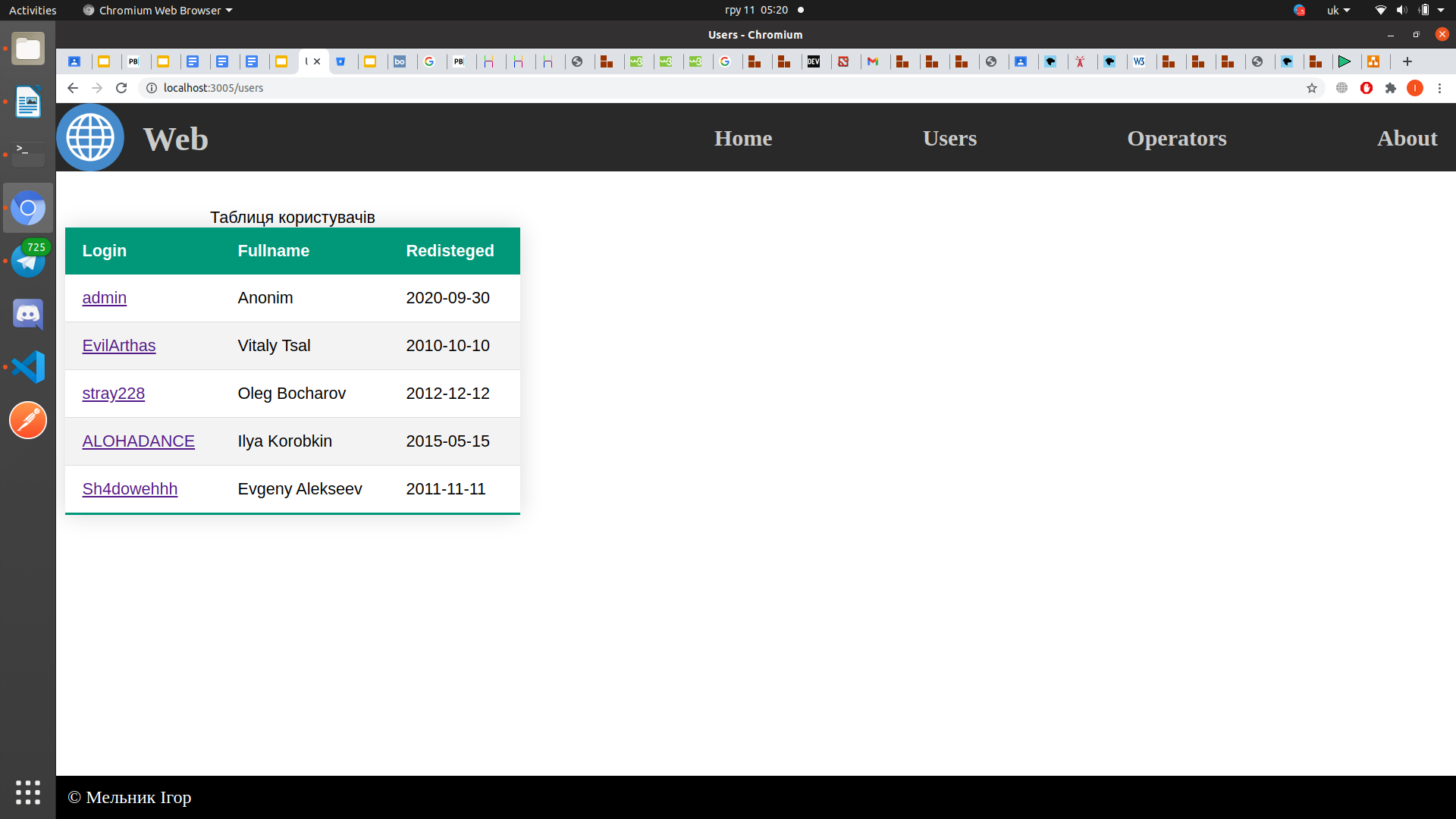
|  |
| --- |
| **views/partials/header.mst** |
| <footer class="footer">  &copy; Мельник Ігор  </footer> |

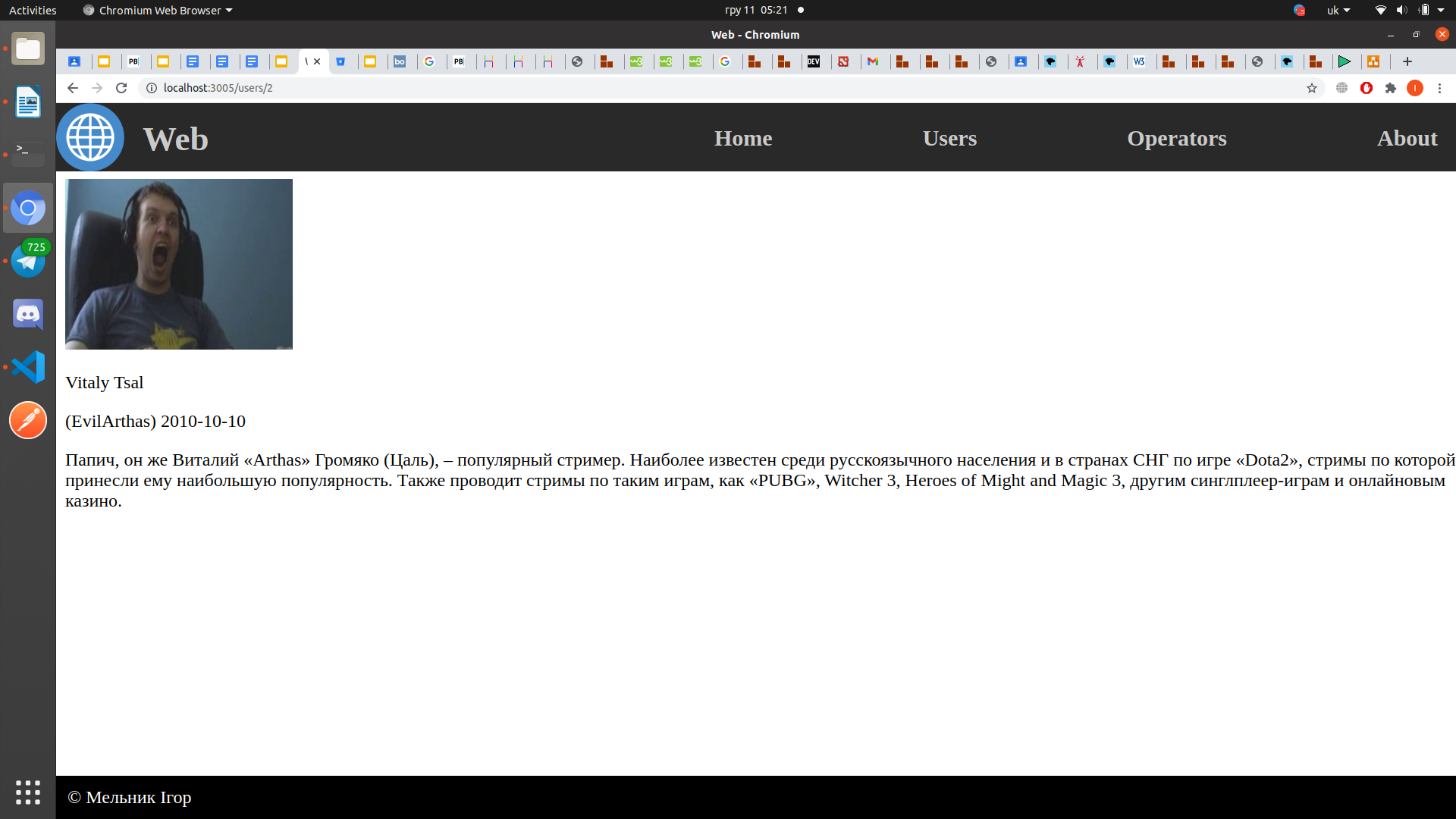
|  |
| --- |
| **public/stylesheets/style.css** |
| .pagination{  float: left;  position: absolute;  left: 0;  top: 230pt;  }  body {  background-color: #ffffff;  background-size: 100%;  }  .header {  list-style-type: none;  margin: 0;  padding: 0;  background-color: rgb(41, 41, 41);  overflow: hidden;  position: absolute;  top: 0;  left: 0;  right: 0;  width: 100%;  height: 60px;  }  .footer {  position: fixed; /\* Фиксированное положение \*/  left: 0; bottom: 0; /\* Левый нижний угол \*/  padding: 10px; /\* Поля вокруг текста \*/  background: #000000; /\* Цвет фона \*/  color: #fff; /\* Цвет текста \*/  width: 100%; /\* Ширина слоя \*/  }  .active {  background-color: #4CAF50;  }  .logo {  float: left;  position: absolute;  left: 20;  }  .logo img {  float: left;  position: relative;  top: 50;  left: 0;  width: 60px;  height: 60px;  }  .logo a {  float: right;  color: #cacaca;  text-align: center;  padding: 14px 16px;  text-decoration: none;  font-size: 30px;  font-weight: bold;  }  .logo a:hover:not(.active) {  background-color: rgb(219, 37, 37);  }  .menu {  float: right;  position: relative;  right: 400;  top: 5px;  }  .menu a {  margin-left: 100px;  float: left;  color: #cacaca;  text-align: center;  padding: 14px 16px;  text-decoration: none;  font-size: 20px;  font-weight: bold;  }  .menu a:hover:not(.active) {  background-color: rgb(219, 37, 37);  }    .styled-table {  border-collapse: collapse;  margin: 25px 0;  font-size: 0.9em;  font-family: sans-serif;  min-width: 400px;  box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);  }  .styled-table thead tr {  background-color: #009879;  color: #ffffff;  text-align: left;  }  .styled-table th,  .styled-table td {  padding: 12px 15px;  }  .styled-table tbody tr {  border-bottom: 1px solid #dddddd;  }  .styled-table tbody tr:nth-of-type(even) {  background-color: #f3f3f3;  }  .styled-table tbody tr:last-of-type {  border-bottom: 2px solid #009879;  }  .content{  top: 50pt;  position: absolute;  }  .circle{  top: 50pt;  position: absolute;  counter-reset: list-counter;  list-style: none;  float:left;  }  .circle li{  margin: 1.5em 0;  }  .circle li:before{  content: counter(list-counter);  counter-increment: list-counter;  width: 1em;  height: 1em;  padding: .5em;  margin-right: 1em;  border-radius: 50%;  border: .25em solid #ccc;  background: #000;  color: #fff;  font-family: arial;  font-weight: bold;  text-align: center;  display: inline-block;  }  .next {  background-color: rgb(41, 41, 41);  /\* Green \*/  border: none;  color: white;  padding: 15px 32px;  text-align: center;  text-decoration: none;  font-size: 16px;  }  .prev {  background-color: rgb(41, 41, 41);  /\* Green \*/  border: none;  color: white;  padding: 15px 32px;  text-align: center;  text-decoration: none;  font-size: 16px;  }  .current {  background-color: rgb(41, 41, 41);  /\* Green \*/  border: none;  color: white;  padding: 15px 32px;  text-align: center;  text-decoration: none;  font-size: 16px;  }  .post {  float: left;  position: absolute;  left: 300pt;  top: 170pt;  background-color: rgb(41, 41, 41);  /\* Green \*/  border: none;  color: white;  padding: 15px 32px;  text-align: center;  text-decoration: none;  font-size: 16px;  } |

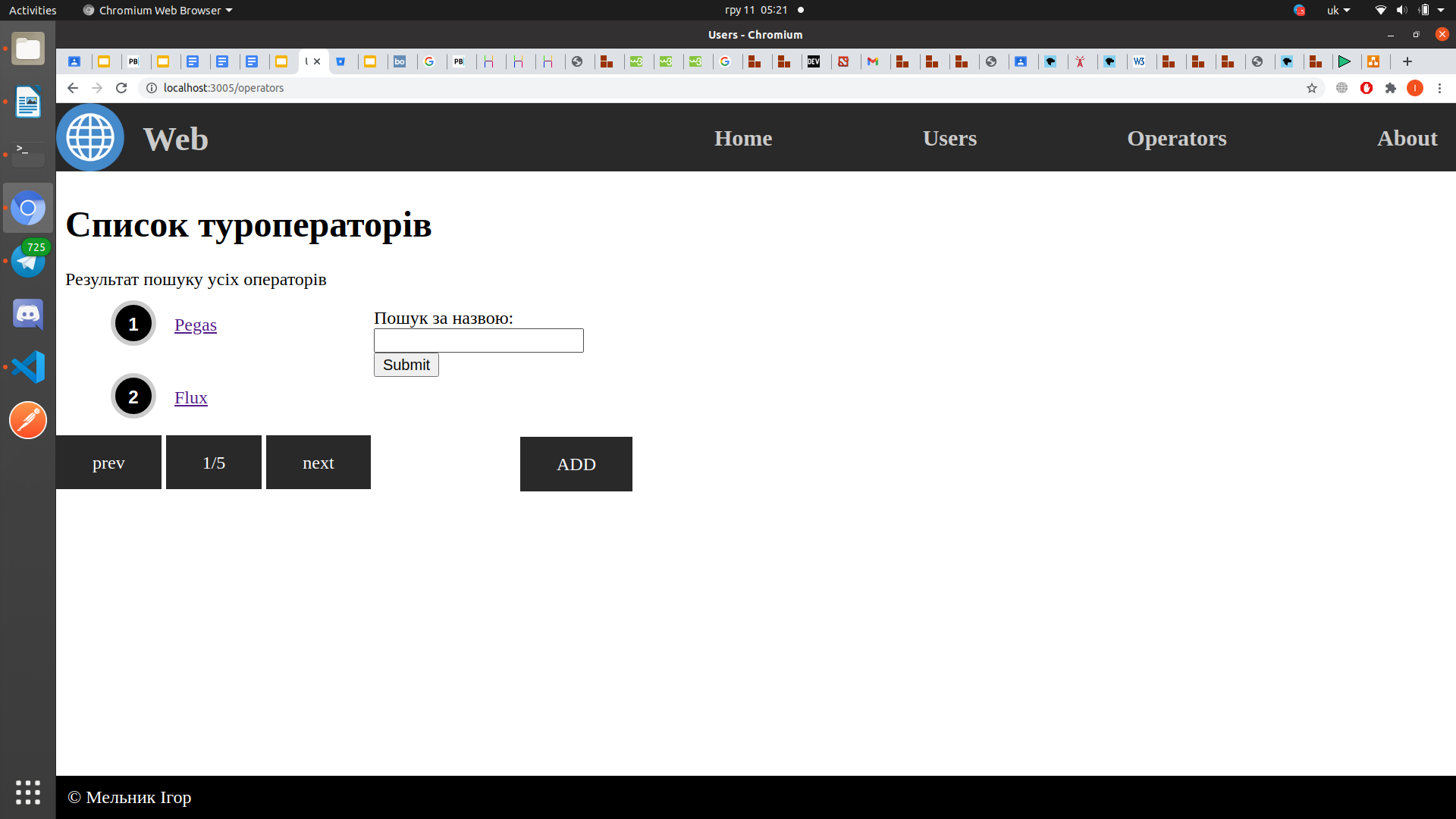
**Зображення усіх розроблених сторінок**

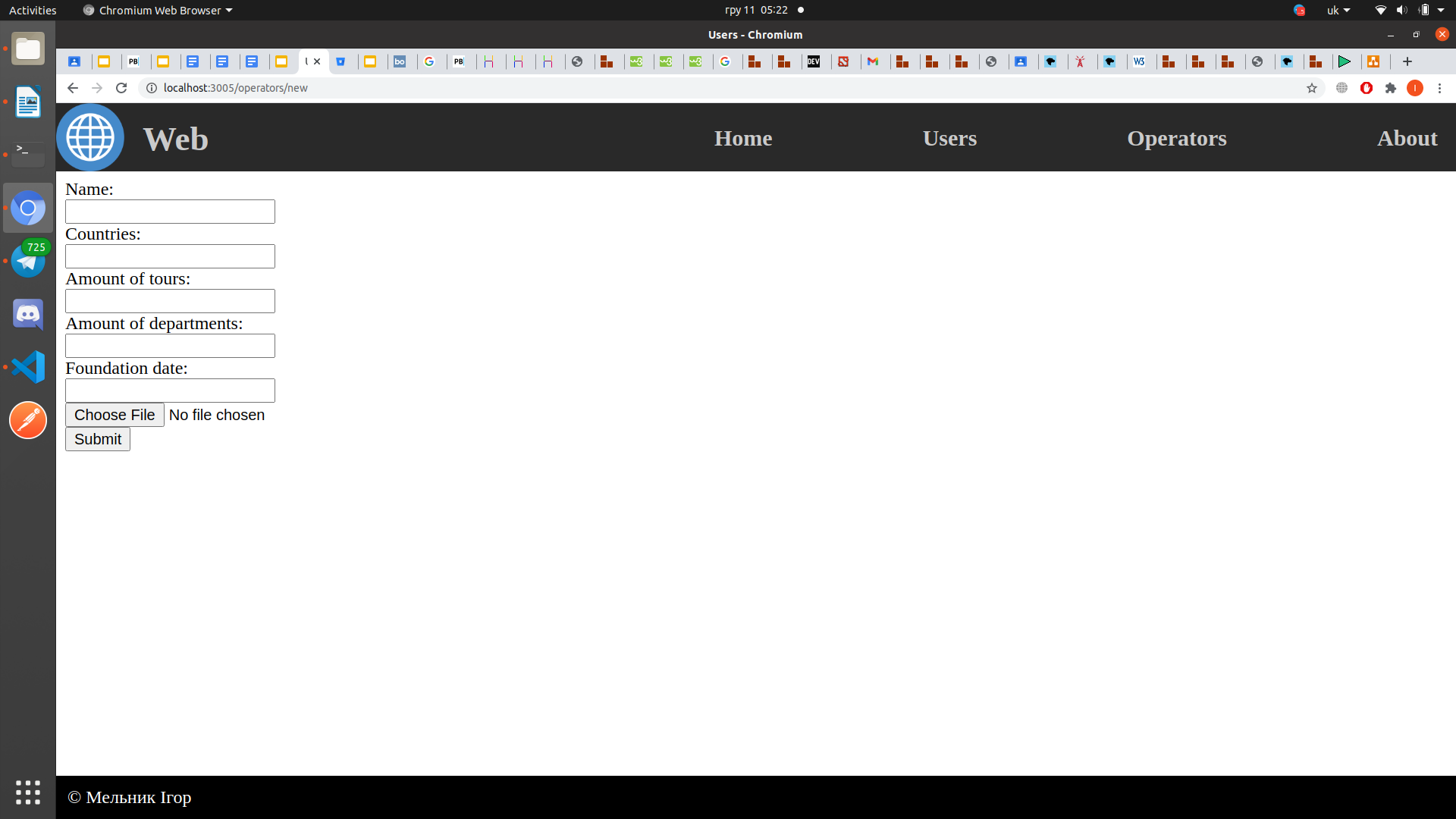


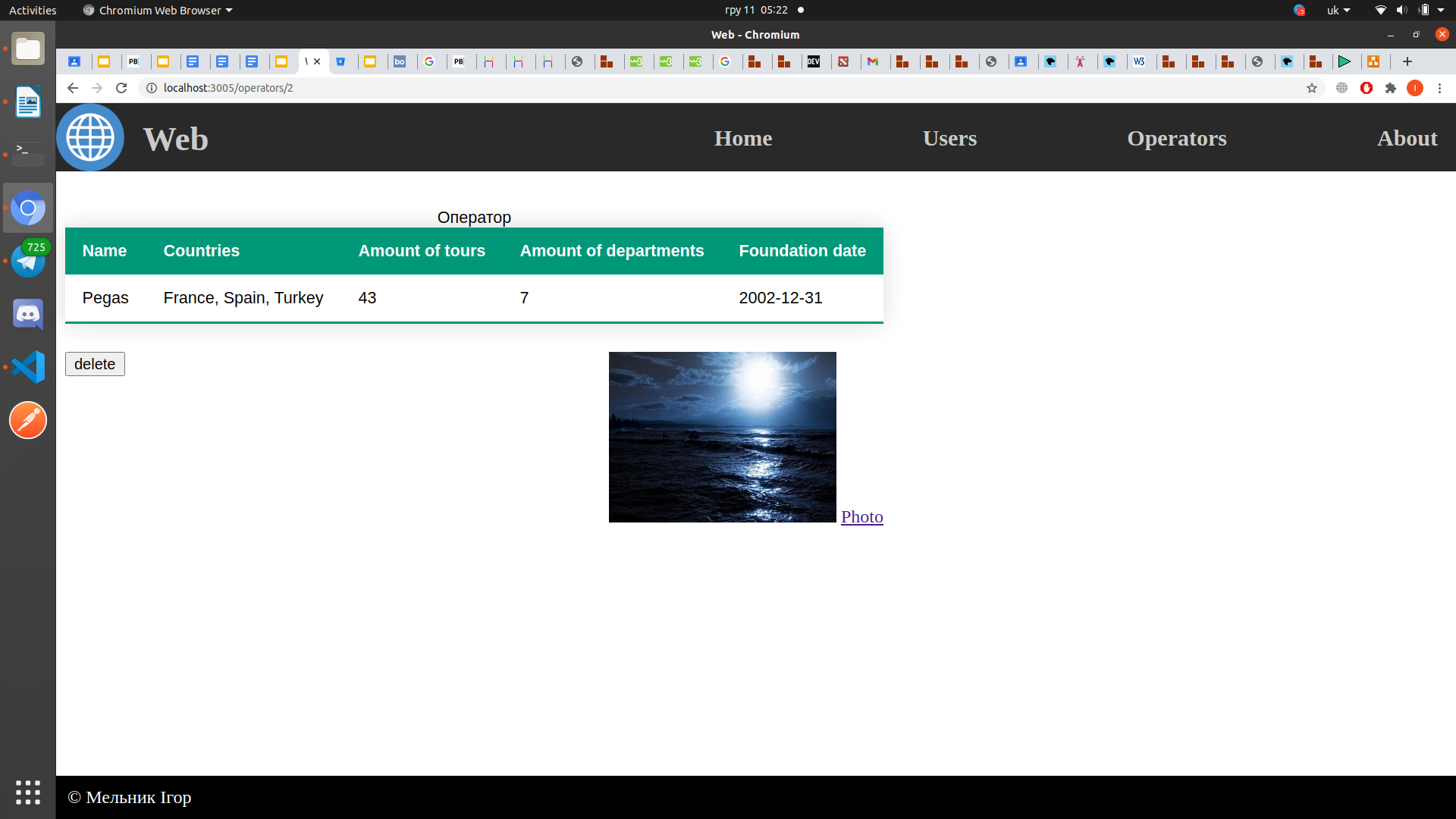
****

****

****

****

****

****

**Висновки**

Виконавши дану лабораторну роботу я навчився створювати веб-сайт, що надає CRUD доступ до ресурсів веб-сервера.  
Ознайомився із шаблонізаторами веб-сторінок для генерації контенту.  
Навчився створювати HTML-форми для взаємодії користувача із веб-сайтом та обробляти на сервері внесені користувачем дані.