Pilnas projekto failų paaiškinimas

# app.py

app.py – Išsamus paaiškinimas eilutė po eilutės

**from flask import Flask, render\_template, request, redirect, url\_for**  
Importuojamos Flask pagrindinės funkcijos: aplikacijos kūrimas, HTML šablonų naudojimas, formų duomenų gavimas, nukreipimas tarp puslapių ir URL generavimas.

**from models import db, Book, Category, User**  
Iš models.py importuojami SQLAlchemy duomenų bazės objektas `db` ir trys modeliai: `Book`, `Category`, `User`.

**from flask\_login import LoginManager, login\_user, logout\_user, login\_required, current\_user**  
Importuojamos `Flask-Login` funkcijos vartotojų prisijungimui, atsijungimui, autentifikacijos tikrinimui ir prisijungusio vartotojo prieigai.

**from werkzeug.security import check\_password\_hash, generate\_password\_hash**  
Importuojamos slaptažodžių šifravimo ir tikrinimo funkcijos.

**import os, sys**  
Importuojami sisteminiai moduliai, reikalingi failų keliams ir .exe veikimui tvarkyti.

**app = Flask(\_\_name\_\_)**  
Sukuriama Flask aplikacija. `\_\_name\_\_` nurodo, kad šis failas yra pagrindinis vykdomas modulis.

**def resource\_path(relative\_path): ...**  
Funkcija leidžia korektiškai gauti failų kelius tiek dirbant su .py, tiek iš .exe failo paleidus aplikaciją.

**db\_path = resource\_path("books.db")**  
Sukuriamas absoliutus kelias iki SQLite duomenų bazės failo.

**app.config['SQLALCHEMY\_DATABASE\_URI'] = f'sqlite:///{db\_path}'**  
Nustatomas duomenų bazės adresas Flask ir SQLAlchemy konfigūracijai.

**app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False**  
Išjungiamas nereikalingas SQLAlchemy resursų stebėjimas, kad mažintų resursų naudojimą.

**app.secret\_key = 'slaptazodis'**  
Slaptas raktas reikalingas seansų ir CSRF apsaugai.

**db.init\_app(app)**  
Inicializuojamas duomenų bazės objektas su Flask aplikacija.

**login\_manager = LoginManager()**  
Sukuriamas login tvarkyklės objektas.

**login\_manager.login\_view = 'login'**  
Nustatomas puslapis, į kurį bus nukreipiama, jei vartotojas bando pasiekti saugomą puslapį neprisijungęs.

**login\_manager.init\_app(app)**  
Inicializuojamas `Flask-Login` su Flask aplikacija.

**@login\_manager.user\_loader**  
Dekoratorius, nurodantis, kaip iš duomenų bazės įkelti vartotoją pagal jo ID.

**def load\_user(user\_id): ...**  
Grąžina `User` objektą pagal vartotojo ID (naudojama `Flask-Login`).

**@app.context\_processor**  
Leidžia perduoti `current\_user` kintamąjį visiems šablonams automatiškai.

**def inject\_user(): ...**  
Funkcija grąžina `current\_user`, kad HTML šablonai visada galėtų naudoti prisijungusio vartotojo informaciją.

**@app.route('/')**  
Pagrindinis puslapio maršrutas.

**def index(): ...**  
Jei vartotojas neprisijungęs, nukreipia į /login. Jei prisijungęs, rodo knygų sąrašą su paieška.

**@app.route('/add', methods=['GET', 'POST'])**  
Maršrutas knygos pridėjimui. Leidžia tiek atvaizduoti formą, tiek ją apdoroti.

**@login\_required**  
Tik prisijungęs vartotojas gali naudotis šiuo puslapiu.

**def add\_book(): ...**  
Rodo formą arba išsaugo naują knygą į duomenų bazę.

**@app.route('/edit/<int:id>', methods=['GET', 'POST'])**  
Redagavimo forma konkrečiai knygai pagal ID.

**def edit\_book(id): ...**  
Rodo redagavimo formą arba atnaujina duomenis duomenų bazėje.

**@app.route('/delete/<int:id>')**  
Ištrina pasirinktą knygą pagal ID.

**def delete\_book(id): ...**  
Suranda knygą, ją ištrina ir grąžina į pagrindinį puslapį.

**@app.route('/login', methods=['GET', 'POST'])**  
Prisijungimo puslapis.

**def login(): ...**  
Tikrina vartotojo vardą ir slaptažodį. Prisijungus – peradresuoja į pagrindinį puslapį.

**@app.route('/logout')**  
Atsijungimo veiksmas.

**def logout(): ...**  
Atsijungia ir grąžina į prisijungimo puslapį.

**@app.route('/change-password', methods=['GET', 'POST'])**  
Slaptažodžio keitimo puslapis.

**def change\_password(): ...**  
Tikrina seną slaptažodį ir, jei jis teisingas, pakeičia į naują.

**if \_\_name\_\_ == '\_\_main\_\_':**  
Užtikrina, kad aplikacija būtų paleista tik jei šis failas yra vykdomas tiesiogiai.

**app.run(debug=True)**  
Paleidžia Flask serverį su `debug` režimu – leidžia matyti klaidas gyvai.

# index.html

index.html – Išsamus paaiškinimas eilutė po eilutės

**<!DOCTYPE html>**  
Nurodo, kad tai HTML5 dokumentas.

**<html lang="lt">**  
Pradeda HTML dokumentą ir nurodo, kad puslapio kalba – lietuvių.

**<head>**  
Dokumento metainformacijos pradžia.

**<meta charset="UTF-8">**  
Nustato simbolių koduotę į UTF-8 – palaikomi lietuviški simboliai.

**<title>Knygų katalogas</title>**  
Puslapio pavadinimas naršyklės skirtuke.

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">**  
Įkeliamas Bootstrap karkasas – leidžia naudoti klases stiliams, pvz., .btn, .container ir t.t.

**<link href="{{ url\_for('static', filename='style.css') }}" rel="stylesheet">**  
Įkeliamas projekto vidinis CSS failas – `style.css` iš `static/` katalogo.

**</head>**  
Baigiasi HTML head blokas.

**<body>**  
Puslapio turinio pradžia.

**<div class="container mt-5">**  
Pagrindinis turinio blokas – Bootstrap `container` su viršutiniu atstumu.

**<div class="hero-section mb-4">**  
Hero paveikslėlio blokas.

**<img src="{{ url\_for('static', filename='objects.jpg') }}" ...>**  
Hero paveikslėlis iš `static/` katalogo.

**</div>**  
Hero paveikslėlio bloko pabaiga.

**<h1 class="mb-4 text-center fw-bold">📚 Knygų katalogas</h1>**  
Pagrindinė puslapio antraštė, su piktograma, centruota ir paryškinta.

**<form method="get" class="d-flex justify-content-center mb-3">**  
Paieškos forma, išdėstyta per centrą.

**<div class="input-group w-75">**  
Paieškos laukelio ir mygtuko grupė (75 % pločio).

**<input type="text" name="search" class="form-control" ...>**  
Paieškos laukas – ieško pagal pavadinimą/autorių.

**<button class="btn btn-outline-primary ms-2" ...>**  
Ieškoti mygtukas – Bootstrap stiliaus su mėlynu apvadu.

**</div>**  
Input grupės pabaiga.

**</form>**  
Formos pabaiga.

**{% if current\_user.is\_authenticated %}**  
Tikrinama ar vartotojas yra prisijungęs.

**<div class="d-flex justify-content-center gap-3 mb-4 flex-wrap">**  
Mygtukų blokas, gražiai centruotas.

**<a href="{{ url\_for('add\_book') }}" class="btn btn-success">➕ Pridėti</a>**  
Pridėti naują knygą.

**<a href="{{ url\_for('change\_password') }}" class="btn btn-warning">🔑 Keisti slaptažodį</a>**  
Slaptažodžio keitimas.

**<a href="{{ url\_for('logout') }}" class="btn btn-danger">🚪 Atsijungti</a>**  
Atsijungimas.

**</div>**  
Mygtukų bloko pabaiga.

**{% endif %}**  
Tikrinimo pabaiga – šie mygtukai matomi tik prisijungus.

**<table class="table table-bordered table-striped">**  
Knygų lentelė – Bootstrap stiliaus su apvadu ir eilutėmis.

**<thead class="table-primary text-center">**  
Lentelės antraštė, su mėlynu fonu.

**<tr> <th>...</th> ... </tr>**  
Lentelės stulpelių pavadinimai.

**</thead>**  
Antraštės pabaiga.

**<tbody>**  
Lentelės turinio pradžia.

**{% for b in books %}**  
Jinja2 ciklas per visas knygas.

**<tr> <td>{{ b.title }}</td> ... </tr>**  
Kiekvienos knygos eilutė lentelėje.

**<td class="text-center"> ... </td>**  
Redagavimo ir trynimo mygtukai.

**{% endfor %}**  
Ciklo pabaiga.

**</tbody>**  
Turinio pabaiga.

**</table>**  
Lentelės pabaiga.

**</div>**  
Pagrindinio konteinerio pabaiga.

**</body>**  
Puslapio pabaiga.

**</html>**  
HTML dokumento pabaiga.

# login.html

login.html – Išsamus paaiškinimas eilutė po eilutės

**<!DOCTYPE html>**  
Nurodo, kad tai HTML5 dokumentas.

**<html lang="lt">**  
HTML dokumentas su nustatyta lietuvių kalba.

**<head>**  
Dokumento metainformacijos pradžia.

**<meta charset="UTF-8">**  
Nustatoma UTF-8 simbolių koduotė – leidžia naudoti lietuviškus simbolius.

**<title>Prisijungimas</title>**  
Nustato naršyklės skirtuko pavadinimą.

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">**  
Importuoja Bootstrap CSS karkasą iš CDN – leidžia naudoti iš anksto apibrėžtus stilius.

**<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">**  
Įkelia vietinį projekto CSS failą (`style.css`), esantį `static/` kataloge.

**</head>**  
Pabaigia head sekciją.

**<body class="login-page">**  
Puslapio turinio pradžia. Klasė `login-page` taikoma specifiniams stiliams.

**<div class="container">**  
Pagrindinis turinio konteineris – centro ribojimui ir stilių taikymui.

**<div class="hero-section mb-4">**  
Hero paveikslėlio blokas su tarpu apačioje.

**<img src="{{ url\_for('static', filename='objects.jpg') }}" ...>**  
Hero paveikslėlis įkeliamas iš `static/` katalogo.

**</div>**  
Hero paveikslėlio bloko pabaiga.

**<div class="card shadow p-4">**  
Kortelės tipo blokas su šešėliu ir padding'u – login forma atrodo kaip atskiras objektas.

**<h2 class="text-center mb-4">🔐 Prisijunkite prie katalogo</h2>**  
Puslapio antraštė – su piktograma ir centruotu tekstu.

**{% if error %}**  
Tikrinama, ar perduotas `error` kintamasis (klaidos pranešimas).

**<div class="alert alert-danger text-center">{{ error }}</div>**  
Klaidos pranešimas, jei vartotojo vardas ar slaptažodis neteisingi.

**{% endif %}**  
Klaidos patikros bloko pabaiga.

**<form method="post">**  
Forma su POST metodu – duomenys siunčiami į serverį saugiai.

**<div class="mb-3">**  
Lauko tarpas apačioje.

**<label class="form-label">Vartotojo vardas</label>**  
Etiketė vartotojo vardo laukui.

**<input type="text" name="username" class="form-control" ...>**  
Įvesties laukas vartotojo vardui.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Antras įvesties laukas.

**<label class="form-label">Slaptažodis</label>**  
Etiketė slaptažodžio laukui.

**<input type="password" name="password" class="form-control" ...>**  
Slaptažodžio įvestis su maskavimu.

**</div>**  
Lauko pabaiga.

**<button type="submit" class="btn btn-primary w-100">Prisijungti</button>**  
Prisijungimo mygtukas – mėlynas, pilno pločio.

**</form>**  
Formos pabaiga.

**</div>**  
Kortelės pabaiga.

**</div>**  
Konteinerio pabaiga.

**</body>**  
Puslapio pabaiga.

**</html>**  
HTML dokumento pabaiga.

# seed.py

seed.py – Išsamus paaiškinimas eilutė po eilutės

**from flask import Flask**  
Importuojamas Flask objektas – reikalingas, kad būtų galima sukurti aplikacijos kontekstą duomenų bazės veiksmams.

**from models import db, Book, Category, User**  
Importuojami duomenų bazės objektas `db` ir visi modeliai – knygos, kategorijos ir vartotojai.

**from werkzeug.security import generate\_password\_hash**  
Importuojama funkcija slaptažodžiui užkoduoti, kad jis būtų saugiai saugomas duomenų bazėje.

**app = Flask(\_\_name\_\_)**  
Sukuriama Flask aplikacija – reikalinga norint naudoti `app.app\_context()`.

**app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///books.db'**  
Nustatomas SQLite duomenų bazės kelias – vietinis failas `books.db`.

**app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False**  
Išjungiamas nereikalingas stebėjimas, kad sutaupytų resursų.

**db.init\_app(app)**  
Priskiriamas Flask aplikacijai duomenų bazės objektas `db`.

**with app.app\_context():**  
Atidaromas Flask aplikacijos kontekstas – būtina sąlyga norint dirbti su duomenų baze už `app.py` ribų.

**db.drop\_all()**  
Ištrinamos visos esamos duomenų bazės lentelės – viskas pradedama nuo nulio.

**db.create\_all()**  
Sukuriamos visos lentelės pagal `models.py` modelius.

**cat1 = Category(name='Romanas')**  
Sukuriama pirmoji kategorija: 'Romanas'.

**cat2 = Category(name='Pasaka')**  
Antra kategorija: 'Pasaka'.

**cat3 = Category(name='Mokslinė literatūra')**  
Trečia kategorija: 'Mokslinė literatūra'.

**db.session.add\_all([cat1, cat2, cat3])**  
Visos kategorijos įkeliamos į `session`.

**db.session.commit()**  
Duomenys įrašomi į duomenų bazę.

**books = [ ... ]**  
Sukuriamas sąrašas knygų objektų su visa būtina informacija: pavadinimas, kaina, autorius, kategorija.

**db.session.bulk\_save\_objects(books)**  
Įrašomos visos knygos į `session` vienu veiksmu – greičiau nei po vieną.

**admin = User(username='admin', password=generate\_password\_hash('admin123'))**  
Sukuriamas administratoriaus vartotojas su saugiai užkoduotu slaptažodžiu.

**db.session.add(admin)**  
Pridedamas vartotojas prie `session`.

**db.session.commit()**  
Visi duomenys galutinai įrašomi į duomenų bazę.

# models.py

models.py – Išsamus paaiškinimas eilutė po eilutės

**from flask\_sqlalchemy import SQLAlchemy**  
Importuojamas SQLAlchemy ORM – objektinis ryšys tarp Python objektų ir SQL lentelių.

**from flask\_login import UserMixin**  
Importuojamas `UserMixin`, kuris prideda `Flask-Login` reikalingas savybes `User` modeliui (pvz. is\_authenticated).

**db = SQLAlchemy()**  
Sukuriamas pagrindinis SQLAlchemy objektas – bus naudojamas visų modelių susiejimui su duomenų baze.

**class User(db.Model, UserMixin):**  
Apibrėžiamas vartotojo modelis – paveldi iš SQLAlchemy modelio ir Flask-Login mixino.

**id = db.Column(db.Integer, primary\_key=True)**  
Unikalus vartotojo ID.

**username = db.Column(db.String(150), unique=True, nullable=False)**  
Unikalus vartotojo vardas (maks. 150 simbolių).

**password = db.Column(db.String(150), nullable=False)**  
Užkoduotas vartotojo slaptažodis (nebūna tuščias).

**class Category(db.Model):**  
Apibrėžiamas kategorijos modelis.

**id = db.Column(db.Integer, primary\_key=True)**  
Unikalus kategorijos ID.

**name = db.Column(db.String(50), nullable=False)**  
Kategorijos pavadinimas (būtinas laukas).

**books = db.relationship('Book', backref='category', lazy=True)**  
Sąsaja: viena kategorija turi daug knygų. Leidžia pasiekti `category.books`.

**class Book(db.Model):**  
Apibrėžiamas knygos modelis.

**id = db.Column(db.Integer, primary\_key=True)**  
Unikalus knygos ID.

**title = db.Column(db.String(100), nullable=False)**  
Knygos pavadinimas.

**price = db.Column(db.Float, nullable=False)**  
Kaina – nurodoma kaip slankusis skaičius.

**release\_date = db.Column(db.String(10))**  
Išleidimo data – saugoma kaip trumpas tekstas (pvz., metai).

**author = db.Column(db.String(100))**  
Autorius – iki 100 simbolių.

**category\_id = db.Column(db.Integer, db.ForeignKey('category.id'), nullable=False)**  
Svetimas raktas į `Category` modelį – nurodo, kokiai kategorijai priklauso knyga.

# style.css

style.css – Išsamus paaiškinimas eilutė po eilutės

**body**  
Pagrindinis viso puslapio fonas ir šriftas.

**background: linear-gradient(to right, #74ebd5, #ACB6E5);**  
Sukuria švelnų perėjimą iš melsvos į šviesiai violetinę – modernus fonas.

**font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;**  
Suteikia švarią, aiškią šriftų eilę – tinka visoms platformoms.

**margin: 0;**  
Pašalina numatytą naršyklės išorinį tarpą.

**padding: 0;**  
Pašalina vidinį tarpą tarp krašto ir turinio.

**.container**  
Pagrindinis kortelės stiliaus blokas.

**background-color: rgba(255, 255, 255, 0.95);**  
Baltas fonas su šiek tiek permatomumo – gražiai išryškėja virš gradientinio fono.

**padding: 2rem;**  
Vidiniai tarpai – atstumas nuo turinio iki kraštų.

**border-radius: 20px;**  
Apvalinti kampai – minkštesnis, šiuolaikiškas dizainas.

**box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);**  
Šešėlis, kad elementas atrodytų pakeltas virš fono.

**margin-top: 2rem;**  
Tarpas nuo viršaus, kad „kortelė“ nebūtų prilipusi prie ekrano viršaus.

**.hero-img**  
Hero paveikslėlio stilius.

**width: 100%;**  
Užima visą konteinerio plotį.

**max-height: 300px;**  
Aukščio ribojimas – neleidžia paveikslui būti per dideliam.

**object-fit: cover;**  
Išlaiko proporcijas ir apkerpa kraštus – neišsitampo.

**border-radius: 20px;**  
Suderina kampų apvalinimą su .container.

**h1, h2**  
Puslapio antraštės.

**text-align: center;**  
Tekstas per vidurį – pabrėžia svarbą.

**font-weight: bold;**  
Storas šriftas – aiškus akcentas.

**color: #333;**  
Tamsiai pilkas tekstas – akiai malonesnis nei grynai juodas.

**.btn**  
Bendras stilius visiems mygtukams.

**border-radius: 20px;**  
Apvalinti mygtukai – modernesnis stilius.

**.btn-success, .btn-primary, .btn-warning, .btn-danger**  
Konkrečių Bootstrap mygtukų tipų individualus paddinimas.

**padding: 6px 12px;**  
Didesnis aktyvus plotas – patogesni paspaudimai.

**.table**  
Lentelės stilius.

**background-color: white;**  
Užtikrina, kad lentelės fonas būtų baltas, net jei kortelė permatoma.

**border-radius: 10px;**  
Apvalinti kampai.

**overflow: hidden;**  
Slepia išlindusius elementus (pvz. šešėlius).

**margin-top: 1rem;**  
Tarpas virš lentelės.

**.alert**  
Bootstrap pranešimų (pvz., login klaidų) atstumas.

**margin-bottom: 1rem;**  
Tarpas po alert'u – kad nesuliptų su forma.

**.d-flex.justify-content-center.gap-3**  
Mygtukų grupės stilius.

**flex-wrap: wrap;**  
Leidžia mygtukams peršokti į naują eilutę, jei netelpa – puiku mobiliesiems.

**body.login-page**  
Login puslapio pagrindinio <body> išdėstymas.

**display: flex;**  
Flex konteineris, kad būtų galima centruoti.

**justify-content: center;**  
Horizontaliai per vidurį.

**align-items: flex-start;**  
Vertikaliai nuo viršaus.

**min-height: 100vh;**  
Viso ekrano aukštis – kad login forma nebūtų prilipusi.

**padding-top: 3rem;**  
Tarpas nuo viršaus – estetinis atskyrimas.

**.login-page .container**  
Login formos konteineris.

**max-width: 500px;**  
Login forma ribojama iki 500px – kad nebūtų per plati ir atrodytų profesionaliai.

# run.py

run.py – Išsamus paaiškinimas eilutė po eilutės

**from app import app**  
Importuojama Flask aplikacija iš `app.py` failo. Tai pagrindinis objektas, kurį paleidžia run.py.

**import webbrowser**  
Importuojamas modulis naršyklės atidarymui – leidžia automatiškai paleisti puslapį naršyklėje.

**import threading**  
Importuojamas `threading` modulis, kad būtų galima paleisti naršyklę asinchroniškai (neblokuojant serverio).

**def open\_browser():**  
Funkcija, kuri atidaro naršyklę į pagrindinį puslapį.

**webbrowser.open\_new("http://127.0.0.1:5000")**  
Atidaro naują naršyklės langą arba kortelę su pagrindiniu puslapio adresu.

**if \_\_name\_\_ == '\_\_main\_\_':**  
Tikrina, ar šis failas paleistas tiesiogiai (ne importuotas).

**threading.Timer(1.5, open\_browser).start()**  
Po 1.5 sekundės paleidžia naršyklės atidarymo funkciją atskirame sraute.

**app.run()**  
Paleidžia Flask serverį. Pagal nutylėjimą – adresu http://127.0.0.1:5000

# change\_password.html

change\_password.html – Išsamus paaiškinimas eilutė po eilutės

**<!DOCTYPE html>**  
Nurodo, kad tai HTML5 dokumentas.

**<html lang="lt">**  
HTML dokumentas su nustatyta lietuvių kalba.

**<head>**  
Pradedama dokumento metainformacijos sekcija.

**<meta charset="UTF-8">**  
Simbolių koduotė nustatoma į UTF-8, palaikanti lietuviškas raides.

**<title>Keisti slaptažodį</title>**  
Nustatomas puslapio pavadinimas naršyklės skirtuke.

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">**  
Pridedamas Bootstrap CSS karkasas – leidžia naudoti greitai stilizuojamus komponentus.

**<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">**  
Įkeliamas vietinis stiliaus failas (style.css) iš Flask static katalogo.

**</head>**  
Uždaro `head` bloką.

**<body>**  
Puslapio turinio pradžia.

**<div class="container py-5">**  
Pagrindinis vidinis blokas su vertikaliu atstumu (padding top/bottom 5).

**<h2 class="text-center mb-4">🔐 Keisti slaptažodį</h2>**  
Puslapio antraštė su piktograma ir centravimu.

**{% if message %}**  
Tikrinama, ar perduotas `message` kintamasis (pranešimas apie sėkmę ar klaidą).

**<div class="alert ...">{{ message }}</div>**  
Pranešimo blokas. Rodomas žalios arba raudonos spalvos pranešimas, priklausomai nuo turinio.

**{% endif %}**  
Pranešimo bloko pabaiga.

**<form method="post">**  
Slaptažodžio keitimo forma su POST metodu – duomenys neatsidengia per URL.

**<div class="mb-3">**  
Formos lauko tarpas.

**<label>Dabartinis slaptažodis</label>**  
Etiketė dabartiniam slaptažodžiui.

**<input type="password" name="current\_password" class="form-control" required>**  
Slaptažodžio laukas su maskavimu.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Naujo slaptažodžio įvedimo laukas.

**<label>Naujas slaptažodis</label>**  
Etiketė naujam slaptažodžiui.

**<input type="password" name="new\_password" class="form-control" required>**  
Naujas slaptažodis – su maskavimu.

**</div>**  
Lauko pabaiga.

**<button type="submit" class="btn btn-primary w-100">💾 Pakeisti</button>**  
Mygtukas – keisti slaptažodį, mėlynas, per visą plotį.

**<a href="{{ url\_for('index') }}" class="btn btn-secondary w-100 mt-2">🔙 Grįžti</a>**  
Papildomas grįžimo mygtukas atgal į katalogą.

**</form>**  
Formos pabaiga.

**</div>**  
Konteinerio pabaiga.

**</body>**  
Puslapio pabaiga.

**</html>**  
HTML dokumento pabaiga.

# add.html

add.html – Išsamus paaiškinimas eilutė po eilutės

**<!DOCTYPE html>**  
HTML5 dokumento pradžia.

**<html lang="lt">**  
Nurodo, kad dokumento kalba yra lietuvių.

**<head>**  
Pradinis metainformacijos blokas.

**<meta charset="UTF-8">**  
UTF-8 simbolių koduotė – leidžia naudoti lietuviškus simbolius.

**<title>Pridėti knygą</title>**  
Naršyklės skirtuko pavadinimas.

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">**  
Pridedamas Bootstrap karkasas iš CDN – leidžia naudoti iš anksto apibrėžtus stilius.

**<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">**  
Įkeliamas vietinis CSS failas – `style.css`, kuris saugomas `static/` kataloge.

**</head>**  
Pabaigiama `head` sekcija.

**<body>**  
Puslapio turinio pradžia.

**<div class="container py-5">**  
Pagrindinis turinio konteineris su vertikaliu padding'u.

**<h2 class="text-success mb-4">➕ Nauja knyga</h2>**  
Puslapio antraštė – žalios spalvos su ikona.

**<form method="post">**  
Forma siunčia duomenis su POST metodu.

**<div class="mb-3">**  
Vienas įvesties laukelis.

**<label class="form-label">Pavadinimas</label>**  
Etiketė laukeliui.

**<input type="text" name="title" class="form-control" required>**  
Tekstinis įvesties laukas knygos pavadinimui.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Kategorijos pasirinkimo laukas.

**<label class="form-label">Kategorija</label>**  
Etiketė kategorijos laukeliui.

**<select name="category\_id" class="form-select" required>**  
Išskleidžiamas sąrašas, kuriame pasirenkama kategorija.

**{% for cat in categories %}**  
Jinja2 ciklas, kuriuo sukuriamos visos kategorijos.

**<option value="{{ cat.id }}">{{ cat.name }}</option>**  
Kiekviena kategorija kaip atskiras pasirinkimas.

**{% endfor %}**  
Ciklo pabaiga.

**</select>**  
Pasirinkimo sąrašo pabaiga.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Kainos įvedimo laukas.

**<label class="form-label">Kaina (€)</label>**  
Etiketė kainai.

**<input type="number" step="0.01" name="price" class="form-control" required>**  
Skaičius su dešimtainiu tikslumu – kainai.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Išleidimo datos įvestis.

**<label class="form-label">Išleidimo data</label>**  
Etiketė.

**<input type="text" name="release\_date" class="form-control">**  
Tekstinis laukas datai (pvz. metams).

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Autoriaus įvedimas.

**<label class="form-label">Autorius</label>**  
Etiketė.

**<input type="text" name="author" class="form-control">**  
Tekstinis laukas autoriaus vardui.

**</div>**  
Lauko pabaiga.

**<button type="submit" class="btn btn-success">💾 Išsaugoti</button>**  
Žalias mygtukas išsaugoti formos duomenims.

**<a href="{{ url\_for('index') }}" class="btn btn-secondary ms-2">🔙 Atgal</a>**  
Grįžimo mygtukas į pagrindinį puslapį.

**</form>**  
Formos pabaiga.

**</div>**  
Pagrindinio konteinerio pabaiga.

**</body>**  
HTML dokumento turinio pabaiga.

**</html>**  
HTML dokumento pabaiga.

# edit.html

edit.html – Išsamus paaiškinimas eilutė po eilutės

**<!DOCTYPE html>**  
HTML5 dokumento pradžia.

**<html lang="lt">**  
Nurodo, kad dokumento kalba yra lietuvių.

**<head>**  
Pradedama HTML metainformacijos sekcija.

**<meta charset="UTF-8">**  
Nustatoma simbolių koduotė – UTF-8, kad veiktų lietuviški simboliai.

**<title>Redaguoti knygą</title>**  
Naršyklės skirtuko pavadinimas.

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">**  
Įkeliama Bootstrap CSS biblioteka iš CDN.

**<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">**  
Įtraukiamas vietinis stilių failas iš Flask `static` katalogo.

**</head>**  
Pabaigiama head sekcija.

**<body>**  
Prasideda puslapio turinys.

**<div class="container py-5">**  
Pagrindinis turinio konteineris su vertikaliu padding'u.

**<h2 class="text-warning mb-4">✏️ Redaguoti knygą</h2>**  
Puslapio antraštė – geltonos spalvos su piktograma.

**<form method="post">**  
Forma, siunčianti duomenis naudojant POST metodą.

**<div class="mb-3">**  
Vienas formos laukas – pavadinimas.

**<label class="form-label">Pavadinimas</label>**  
Etiketė tekstiniam laukeliui.

**<input type="text" name="title" class="form-control" value="{{ book.title }}" required>**  
Laukas pavadinimui su iš anksto įrašyta reikšme.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Kategorijos pasirinkimo laukas.

**<label class="form-label">Kategorija</label>**  
Etiketė.

**<select name="category\_id" class="form-select" required>**  
Išskleidžiamas sąrašas kategorijai.

**{% for cat in categories %}**  
Ciklas per visas galimas kategorijas.

**<option value="{{ cat.id }}" {% if cat.id == book.category\_id %}selected{% endif %}>{{ cat.name }}</option>**  
Sąlyginai parenkama esama knygos kategorija.

**{% endfor %}**  
Ciklo pabaiga.

**</select>**  
Select lauko pabaiga.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Kainos laukas.

**<label class="form-label">Kaina (€)</label>**  
Etiketė.

**<input type="number" step="0.01" name="price" class="form-control" value="{{ book.price }}" required>**  
Skaičiaus įvestis su iš anksto įrašyta verte.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Išleidimo data.

**<label class="form-label">Išleidimo data</label>**  
Etiketė.

**<input type="text" name="release\_date" class="form-control" value="{{ book.release\_date }}">**  
Tekstinis laukas su iš anksto įrašyta knygos išleidimo data.

**</div>**  
Lauko pabaiga.

**<div class="mb-3">**  
Autorius.

**<label class="form-label">Autorius</label>**  
Etiketė.

**<input type="text" name="author" class="form-control" value="{{ book.author }}">**  
Tekstinis laukas su iš anksto įrašytu autoriumi.

**</div>**  
Lauko pabaiga.

**<button type="submit" class="btn btn-warning">💾 Atnaujinti</button>**  
Mygtukas formos išsaugojimui.

**<a href="{{ url\_for('index') }}" class="btn btn-secondary ms-2">🔙 Atgal</a>**  
Mygtukas grįžti atgal.

**</form>**  
Formos pabaiga.

**</div>**  
Konteinerio pabaiga.

**</body>**  
Puslapio pabaiga.

**</html>**  
HTML dokumento pabaiga.