## ПЛАН НА ДИПЛОМНИЯ ПРОЕКТ

**Тема:** Удостоверяване на самоличността и оторизация в уеб приложения

* **I. Теоретична част – (около 12–14 стр.)**
* **1. Основни понятия и принципи в уеб сигурността (3 стр.)**
* Понятието за удостоверяване (authentication) и оторизация (authorization)
* Значение на сигурността в съвременните уеб приложения
* **2. Удостоверяване на самоличността (authenticaton) (3 стр.)**
* Основни подходи за удостоверяване: еднофакторно, двуфакторно и многофакторно удостоверяване
* Механизми и протоколи: потребителско име и парола, сертификати, OAuth/SSO
* Типични уязвимости и добри практики
* **3. Оторизация (authorization) (3 стр.)**
* Понятие за роля, право и политика за достъп
* Видове оторизация: по роли (Role-Based Access Control – RBAC), по атрибути (Attribute-Based Access Control – ABAC)
* Примери за различни нива на достъп в приложения
* **4. Архитектурни модели и стандарти в уеб удостоверяването (3 стр.)**
* Сесии, бисквитки (cookies), токени (JWT)
* Инструменти и библиотеки: сравнение между различни подходи и технологии (с фокус върху Python и Django)
* **5. Средства за удостоверяване и оторизация в Django (4–5 стр.)**
* Разглеждане на вградената автентикация в Django (django.contrib.auth)
* Моделът на потребителя (User model), групи и права
* Готови изгледи (views) и форми за логин, регистрация, смяна на парола
* Разширяване и персонализация на логиката за управление на достъпа
* **II. Практическа част**
* **6. Обща архитектура и постановка на приложението (2–3 стр.)**
* Описание на целите – реализиране на идентификация, регистрация и оторизация
* Основни файлове във функционалната част: models.py, forms.py, views.py, urls.py, admin.py
* **7. Реализация на моделите за потребителски данни (3 стр.)**
* Описание на използвания или разширения User модел
* Създаване и организация на роли, групи, права (групи с права за достъп до ресурси)
* Административни инструменти (admin.py): възможности за преглед и редакция
* **8. Реализация на формите за регистрация и вход (3 стр.)**
* Описание на forms.py (класове за login, регистрация, редакция на профил)
* Валидация и обработка на данни във формите
* Изглед на формите на потребителския интерфейс (екрани, полета, съобщения)
* **9. Реализация на изгледите за удостоверяване и управление на профил (3 стр.)**
* Изгледи за вход и изход (login/logout)
* Изглед за регистрация на нов потребител
* Изглед за преглед и редакция на профилните данни
* Обработка на сесии, пренасочвания и съобщения
* **10. Оторизация и управление на достъпа (3 стр.)**
* Проверка за идентифициран потребител (декоратори login\_required, permission\_required и т.н.)
* Симулиране на роли и пример с ограничения в левия панел на интерфейса
* Примери за съдържание достъпно само за влезли или само за потребители с определени права
* **11. Представяне на потребителския интерфейс (2 стр.)**
* Демонстриране на дизайн: login екран, регистрационна форма, профил, ляв панел с информация
* Примери за различни състояния на интерфейса: незаконосъобразен достъп, съобщения при грешка, успешна регистрация
* **12. Тестване на системата и демонстрация (1.5–2 стр.)**
* Примерни сценарии за вход, регистрация, промяна на данни
* Скрийншоти и описание на протичането
* Оценка на сигурността и препоръки при реална употреба
* **III. Приложения – (2–4 стр.)**
* Основни извадки от кода: models.py, forms.py, views.py, urls.py, admin.py
* Примерни снимки на интерфейса
* Примерни права и структури на групи
* Диаграми и допълнителни обяснения при необходимост

**Общи бележки:**

* Всеки раздел от плана може допълнително да бъде разширен с примери от изходния код, таблици, диаграми и сравнения.
* Изискването за минимум 30 страници ще бъде постигнато при стандартна академична структура и форматиране, като се изпишат на разбираем и структуриран език всички описани точки, с подробни примери, обяснения и илюстрации.