

**Dizajni I Sistemit te Softuerit**

Viti akademik 2024 – 2025

**Emri I Projektit: Luxury Step**

Aid Aliu

Aldin Shabani

Blend Kqiku

Profesori: PhD. Cand. Greta Ahma

GitHub Link: https://github.com/AidAliu/LuxuryStep

Prill 2025

**Permbajtja**

[Luxury Step 2](#_Toc187702145)

[1 Hyrje 2](#_Toc187702146)

[2 Conceptual Model 3](#_Toc187702147)

[2.1 Conceptual Model-Presentation Layer 3](#_Toc187702147)

[2.2 Conceptual Model-Business Logic Layer 3](#_Toc187702147)

[2.3 Conceptual Model- Data Access Layer 3](#_Toc187702147)

[3.Utility Tree 5](#_Toc187702151)

[3.1 Usage (Përdorimi) 6](#_Toc187702152)

[3.2 Performance (Performancë) 6](#_Toc187702153)

[3.3 Reliability (Besueshmëri) 6](#_Toc187702154)

[3.4 Modifiability (Modifikueshmëri) 6](#_Toc187702155)

[3.5 Reusability(Ripërdorimi) 7](#_Toc187702156)

[3.6 Adaptability(Përshtatja) 7](#_Toc187702156)

[3.7 Security(Siguria) 7](#_Toc187702156)

[4. Product Overview 7](#_Toc187702157)

[4.1 Product Vision 7](#_Toc187702158)

[4.2. Key Features 7](#_Toc187702159)

[4.3 Stakeholders 8](#_Toc187702160)

[4.4 Target Market 8](#_Toc187702161)

[5. Architectural Models 10](#_Toc187702164)

[Fig 2. RidePlanner Layer Overview 10](#_Toc187702165)

[5.1 Layer Responsibilities 11](#_Toc187702166)

[5.2 Presentation Layer Decomposition 11](#_Toc187702167)

[5.2.1 Presentation Layer Responsibilities 12](#_Toc187702168)

[5.2.2 Model Layer Interface Specifications 13](#_Toc187702169)

[5.2.3 Presentation Layer Design Rationale 14](#_Toc187702170)

[5.3 Bussines Logic Layer Decomposition 14](#_Toc187702171)

[5.3.1 Bussines Logic Layer Responsibilities 16](#_Toc187702172)

[5.3.2 Bussines Logic Layer Interface Specifications 17](#_Toc187702173)

[5.3.3 Bussines Logic Layer Design Rationale 19](#_Toc187702174)

[5.4Data Access Layer Decomposition 20](#_Toc187702175)

[5.4.1 Data Access Layer Responsibilities 21](#_Toc187702176)

[5.4.2 Data Access Layer Specifications 22](#_Toc187702177)

[5.4.3 Data Access Layer Design Rationale 23](#_Toc187702178)

[5.5Infrastructure Layer Decomposition 24](#_Toc187702179)

[5.5.1 Infrastructure Layer Responsibilities 25](#_Toc187702180)

[5.5.2 Infrastructure Layer Specifications 26](#_Toc187702181)

[5.5.3 Infrastructure Layer Design Rationale 27](#_Toc187702182)

[5.6 Runtime Components 28](#_Toc187702183)

[6.6 Mapping Between Models 29](#_Toc187702184)

[5.7 Architectural Design Rationale 29](#_Toc187702185)

[6.Ride Planner Detailed Desing Document 31](#_Toc187702186)

[6.1 Mid Level Desing Models 31](#_Toc187702187)

[6.1.1 User Modules Responsibilities 32](#_Toc187702188)

[6.1.2 User Module Specifications 33](#_Toc187702189)

[6.2.1 Bus Schedule Module Responsibilities 34](#_Toc187702190)

[5.2.2 Bus Schedule Module Specifications 35](#_Toc187702191)

[5.2.2 Bus Schedule Module Specifications 36](#_Toc187702192)

[5.3.1 TaxiReservation Modules Responsibilities 38](#_Toc187702193)

[5.3.2 TaxiReservation Module Specifications 39](#_Toc187702194)

[5.4.1 TaxiDriverDashboard Modules Responsibilities 39](#_Toc187702195)

[5.4.2 TaxiDriverDashboard Module Specifications 40](#_Toc187702196)

[6.5 Low-Level Desing 42](#_Toc187702197)

[6.5.2 Detailed View of the Modules of Low Levels 43](#_Toc187702198)

[6.6 Mapping Beetwen Models 44](#_Toc187702199)

[6.7 Detailed Desing Rationale 45](#_Toc187702200)

[7 Diagramet 46](#_Toc187702201)

[7.1 Activity Diagrams 46](#_Toc187702202)

[7.1.1 User Authentication 46](#_Toc187702203)

[7.1.2 Taxi Reservation 48](#_Toc187702204)

[7.2 Sequence Diagrams 51](#_Toc187702205)

[7.2.1 User Authentication 51](#_Toc187702206)

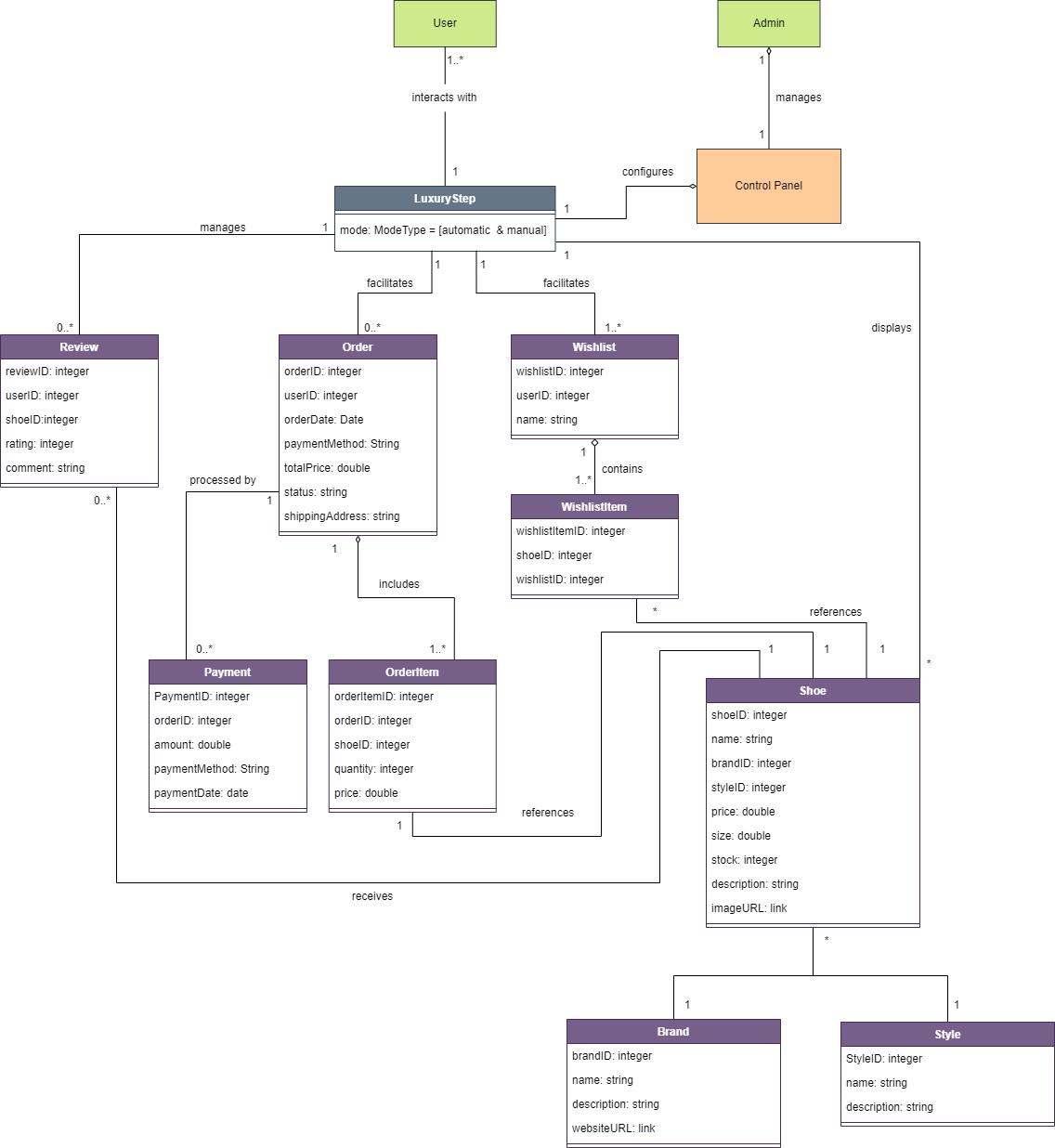
[7.2.2 Taxi Reservation 52](#_Toc187702207)

1. **Hyrje**

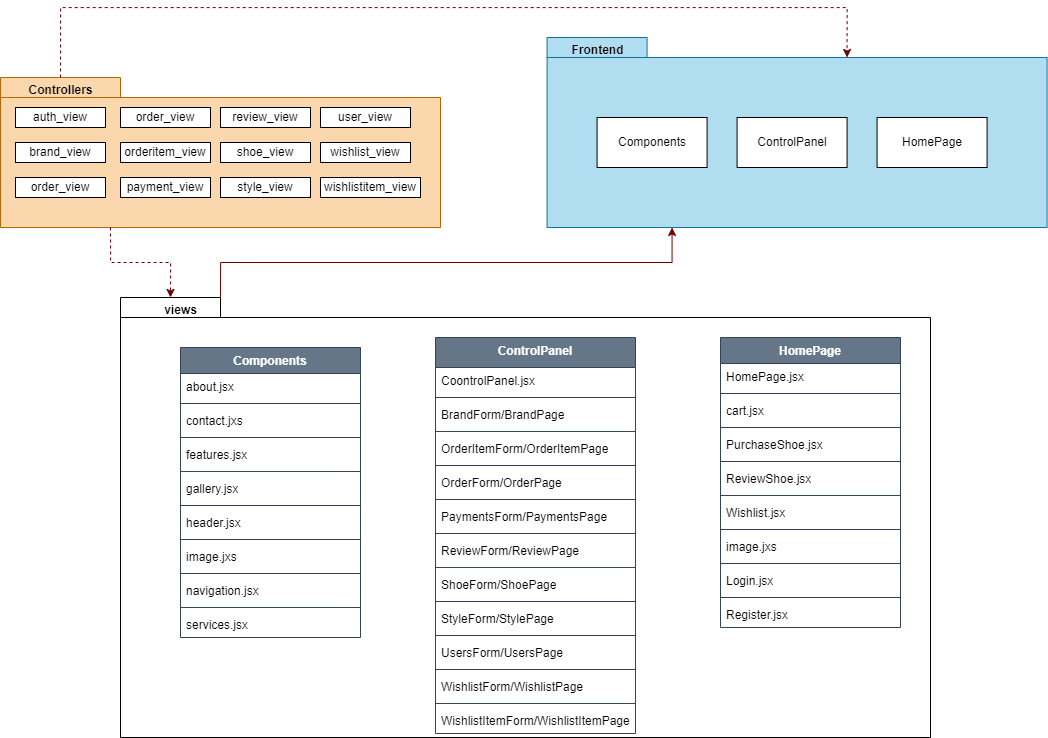
**LuxuryStep** është një platformë inovative që kombinon fuqinë e teknologjisë me pasionin për këpucët luksoze, duke krijuar një hapësirë ku përdoruesit mund të eksplorojnë, rishikojnë dhe blejnë produkte me cilësi të lartë. E ndërtuar me **Django** për backend dhe **React** për frontend, **LuxuryStep** synon të ofrojë një eksperiencë të shpejtë, të sigurt dhe të personalizuar për të gjithë adhuruesit e modës dhe koleksionistët e këpucëve. Platforma shërben si një vendtakim për entuziastët e këpucëve, duke ofruar mundësinë për të ndarë mendimet e tyre për marka dhe modele të ndryshme, si dhe për të bërë blerje të sigurta nga shitës të verifikuar.

**LuxuryStep** është konceptuar si një ekosistem i plotë për komunitetin e këpucëve luksoze, duke pasur synime të qarta si:

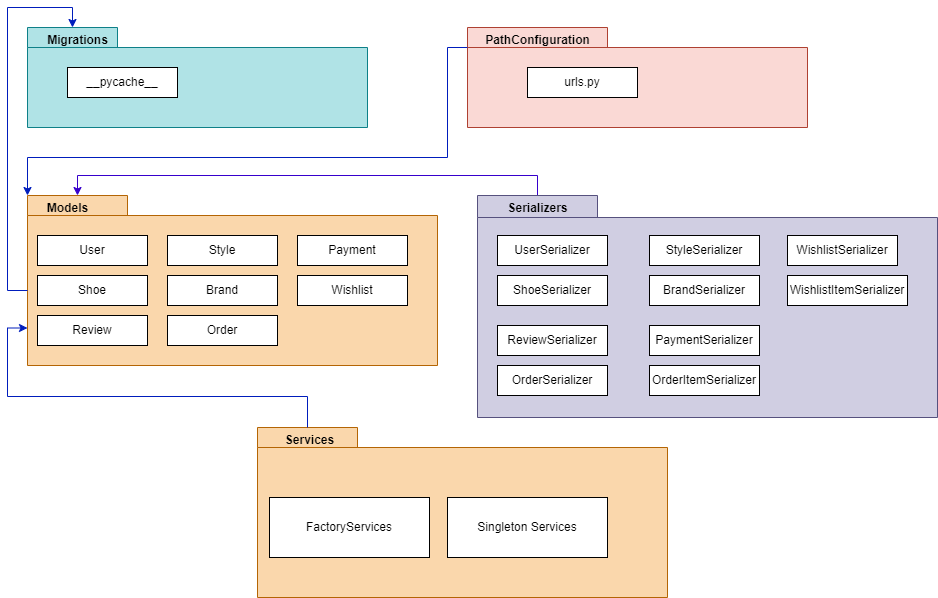
1. **Lehtësimi i Zgjedhjes për Blerësit** – Përdoruesit mund të lexojnë rishikime autentike dhe të bazuara në eksperiencën e konsumatorëve të tjerë para se të marrin një vendim për blerje.
2. **Rritja e Besimit përmes Transparencës** – Çdo produkt ka një vlerësim të detajuar dhe rishikime të verifikuara për të garantuar autenticitetin dhe cilësinë.
3. **Një Hapësirë për Adhuruesit e Këpucëve** – LuxuryStep krijon një komunitet ku koleksionistët dhe të apasionuarit pas këpucëve mund të ndajnë mendimet e tyre.
4. **Menaxhimi i Preferencave Personale** – Përdoruesit mund të krijojnë një listë dëshirash me këpucët që pëlqejnë dhe të ndjekin uljet e çmimeve.
5. **Sigurimi i Transaksioneve të Besueshme** – Platforma mundëson blerjen dhe shitjen e këpucëve përmes një sistemi të sigurt pagese dhe verifikimi.
6. **Conceptual Model**

****

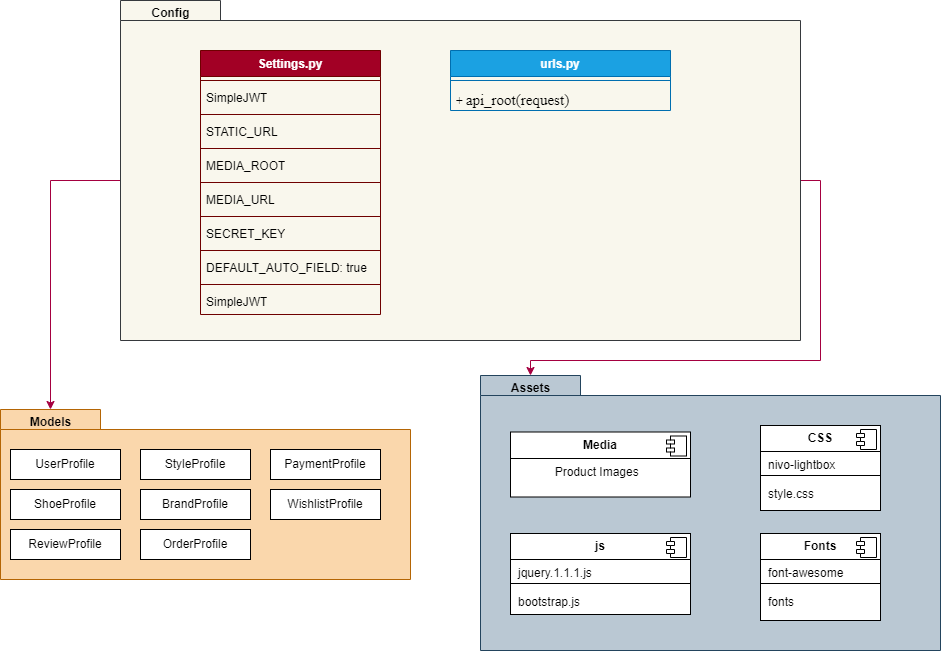
* 1. **Conceptual Model-Presentation Layer**

****

* 1. **Conceptual Model-Business Logic Layer**

****

* 1. **Conceptual Model- Data Access Layer**

****

1. **Utility Tree**

**3.1 Usage**

* **Shkruaj Rishikime**: Lejo përdoruesit të japin vlerësime për këpucët e blera.
* **Blerja e Këpucëve**: Siguro një proces të lehtë dhe të sigurt për blerjen dhe pagesën.
* **Menaxhimi i Listës së Dëshirave**: Mundëso shtimin ose heqjen e këpucëve nga lista e dëshirave.

**3.2 Performance**

* **Ngarkim i Shpejtë i Faqeve: Optimizimi i kohës së ngarkimit për kërkimet dhe faqet e produkteve.**

**3.3 Reliability**

* **Transaksione të Sigurta**: Sigurimi i metodave të pagesës të koduara dhe të mbrojtura.

**3.4 Modifiability**

* **Personalizimi i Ndërfaqes**: Lejo administratorët të modifikojnë ndërfaqen për promovime.

**3.5 Reusability**

* **API për Produktet**: Përdorimi i API-ve për të integruar sistemet e jashtme për përditësimin e inventarit ose rekomandime.
* **Analiza e Rishikimeve**: Ofrimi i përmbledhjeve nga vlerësimet e përdoruesve për qëllime marketingu.

**3.6 Adaptability**

* **Mbështetje për Monedhën**: Lejo platformën të trajtojë pagesa me kartë (VISA/MASTER), PayPal ose para në dorë me dollarë.
* **Infrastrukturë e Shkallëzueshme**: Sigurimi që sistemi të përballojë rritjen e përdoruesve dhe inventarit.

**3.7 Security**

* **Sistem i Autentifikimit**: Siguro hyrje të mbrojtura me enkriptim të fjalëkalimeve dhe role të përdoruesve.
* **Ndjekja e Porosive**: Menaxhimi dhe shfaqja e historikut të porosive në mënyrë të sigurt.

1. **Product Overview**

**4.1 Product Vision**

LuxuryStep është një platformë e specializuar e-commerce e krijuar për të ofruar një përvojë të lehtë dhe të përshtatshme për adhuruesit dhe blerësit e këpucëve. Ajo kombinon funksionalitetin e një platforme tradicionale të tregtisë elektronike me një sistem vlerësimi dhe rishikimi, duke i lejuar përdoruesit të shfletojnë, blejnë dhe rishikojnë lloje të ndryshme të këpucëve. Platforma gjithashtu mbështet menaxhimin e listave të dëshirave dhe ofron kontrolle administrative për moderimin e rishikimeve të këpucëve.

**4.2 Key Features:**

 **Katalog i Gjithanshëm i Këpucëve:**

* Ofron një gamë të gjerë këpucësh të kategorizuara sipas markës, stilit, madhësisë dhe çmimit.
* Përshkrime të detajuara, informacione mbi stokun dhe imazhe me cilësi të lartë për çdo produkt.

 **Menaxhimi i Listës së Dëshirave:**

* Përdoruesit mund të krijojnë dhe menaxhojnë lista dëshirash për këpucët e preferuara.
* Mundësia për të ruajtur këpucët për blerje ose referencë të mëvonshme.

 **Porositja dhe Gjurmimi i Porosive:**

* Lejon përdoruesit të shtojnë këpucët në shportë, të vendosin porosi dhe të ndjekin statusin e tyre.
* Përfshin azhurnime në kohë reale mbi stokun dhe llogaritje dinamike të çmimeve.

 **Rishikimet e Përdoruesve:**

* Përdoruesit mund të vlerësojnë dhe të shkruajnë rishikime për këpucët e blera.
* Ndihmon përdoruesit e tjerë të marrin vendime të informuara bazuar në vlerësimet dhe komentet.

 **Paneli Administrativ:**

* Lejon administratorët të moderojnë rishikimet e këpucëve dhe të ruajnë integritetin e sistemit të rishikimeve.
* Përfshin funksionalitete për menaxhimin e produkteve dhe komenteve të përdoruesve.

**4.3 Stakeholders:**

 **Përdoruesit Finalë:**

* Përdoruesit kryesorë që shfletojnë, blejnë dhe rishikojnë këpucët.
* Krijojnë lista dëshirash të personalizuara dhe vendosin porosi.

 **Markat e Këpucëve:**

* Partnerë që promovojnë produktet e tyre në platformë.
* Përfitojnë nga rishikimet e përdoruesve dhe rritja e dukshmërisë.

 **Administratorët e Platformës:**

* Mbikëqyrin dhe menaxhojnë ndërveprimet e përdoruesve, rishikimet dhe produktet.

 **Zhvilluesit:**

* Mirëmbajnë platformën dhe përmirësojnë funksionalitetet e saj për shkallëzueshmëri dhe përdorshmëri më të mirë.

**4.4 Target Market:**

 **Dizajn Dinamik i Bazës së të Dhënave:**

* Lidhjet midis përdoruesve, rishikimeve, porosive dhe këpucëve sigurojnë një menaxhim të qëndrueshëm dhe të besueshëm të të dhënave.
* Arkitekturë modulare që lejon shkallëzueshmëri dhe mirëmbajtje të lehtë.

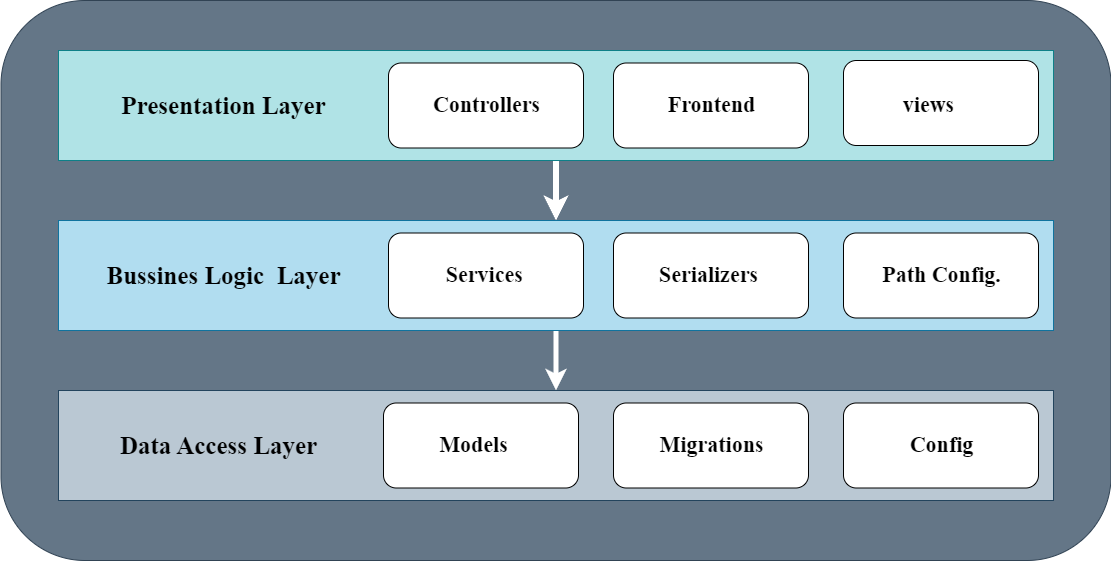
 **Ndërfaqe Interaktive për Përdoruesit:**

* E dizajnuar për të qenë e përgjegjshme dhe intuitive, duke ofruar një përvojë të qetë të shfletimit dhe blerjes.
* Integrimi i teknologjive moderne të uebit për performancë të përmirësuar.

 **Transaksione të Sigurta:**

* Përdor porta të sigurta pagese dhe ruajtje të dhënash të koduara për të mbrojtur informacionin e përdoruesve.

1. **Architectural Models**



Platforma **LuxuryStep** adopton arkitekturën **Model-View-Template (MVT)**, e lehtësuar nga **Django REST Framework**. Kjo arkitekturë siguron ndarjen e përgjegjësive, duke mundësuar modularitet dhe mirëmbajtje më të mirë:

* **Modeli**: Përfaqëson skemën e bazës së të dhënave dhe trajton logjikën e biznesit, duke përfshirë entitete si **Shoe, Review, Order** dhe **Wishlist**.
* **View (Pamja)**: Proceson kërkesat hyrëse, ndërvepron me modelet dhe përgatit përgjigjet.
* **Template (Shablloni)**: Menaxhon **Presantation Layer**, duke ofruar përmbajtje dinamike të renderuar në frontend.

Kjo arkitekturë krijon kufij të qartë midis menaxhimit të të dhënave, logjikës së biznesit dhe ndërfaqes së përdoruesit, duke ofruar fleksibilitet dhe shkallëzueshmëri.

**MVT** u zgjodh sepse:

* Integrohet pa probleme me **Django REST Framework** për ndërtimin e API-ve të fuqishme.
* Mundëson zhvillim të shpejtë duke përdorur komponentë të ripërdorshëm dhe konvencione.

## **5.1** **Layer Responsibilities**

|  |  |
| --- | --- |
| **Layer** | **Përshkrimi** |

|  |  |
| --- | --- |
| **Presentation Layer** | Shtresa që menaxhon ndërfaqen e përdoruesit dhe ndërveprimin me aplikacionin. Në **LuxuryStep**, kjo shtresë përfshin **frontend-in** (React), **views** (Django REST Framework) dhe **controllers** që përpunojnë kërkesat e përdoruesit. |

|  |  |
| --- | --- |
| **Business Logic Layer** | Përgjegjëse për rregullat e biznesit dhe logjikën e aplikacionit. Kjo shtresë përmban **services**, **serializers** dhe **path config** (urls.py që menaxhon rruget e API-së). |

|  |  |
| --- | --- |
| **Data Access Layer** | Merret me ruajtjen dhe menaxhimin e të dhënave në bazën e të dhënave. Përfshin **models**, **migrations** dhe **config** (settings.py ku përcaktohen lidhjet me bazën e të dhënave). |

**5.2 Presentation-Layer Decomposition**

**Controllers**: Përmban funksionet **API** që trajtojnë kërkesat nga klienti dhe ndërveprojnë me shtresën e logjikës së biznesit. Këto janë zakonisht **View** ose **APIViews** në Django REST Framework.

**Frontend**: Aplikacioni **React** që konsumon API-të e Django dhe paraqet UI-në për përdoruesit. Përfshin komponentët **React** dhe ruterin për navigim.

**Views**: Në kontekstin e **Django-s**, përfshin klasat dhe funksionet në **views folder** që trajtojnë logjikën e kërkesave **HTTP** dhe kthejnë përgjigjet. Në **React**, kjo mund të jetë komponentët që menaxhojnë interaktivitetin.

**Controllers**

* **ShoeView →** Menaxhon kërkesat për menaxhimin e këpucëve (CRUD).
* **ReviewView →** Lejon krijimin dhe shfaqjen e rishikimeve për këpucët.
* **OrderView →** Trajton porositë e bëra nga përdoruesit.

**Frontend (React)**

* **Wishlist.jsx →** Shfaq listën e këpucëve që janë në wishlist.
* **Purchase.jsx →** Paraqet faqen ku kryhet pagesa.
* **Cart.jsx →** Menaxhon shportën e blerjeve dhe përfundimin e porosive.

**Views (Django)**

* **ShoeAPIView →** Merr një listë këpucësh ose detajet e një këpuce.

**ViewModels(Django Serializers):**

Në kontekstin e Django-s, Serializers shërbejnë si një formë e ViewModel që përgatit dhe validon të dhënat përpara se të kthehen në klient ose të ruhen në bazën e të dhënave. Ato mundësojnë ndarjen e logjikës së prezantimit nga modelet e të dhënave.

**Rrjedhja e Nderveprimit:**

1. Perdoruesi ndervepron me nje component ne React si ‘Cart.jsx.
2. Ky component degron nje kerkese HTTP ne nje API te Django, p.sh., ‘OrderView’.
3. ‘OrderView’ komunikon me logjiken e biznesit dhe perdore ‘OrderSerializer’ per te perpunuar te dhenat.
4. Pergjigja JSON e serializuar dergohet mbrapsht ne frontend per t’u paraqitur ne UI.

**Perputhja me Modelin Konceptual:**

* ‘ShoeView’ lidhet me entitetin ‘Shoe’.
* ‘ReviewView’ menaxhon entitetin ‘Review’.
* ‘OrderView’ perfaqeson nderveprimet me ‘Order’ dhe ‘OrderItem’.
* Komponentet si ‘Wishlist.jsx’ lidhen direkt me ‘Wishlist’ dhe ‘WishlistItem’ ne modelin konceptual.