**EcoCart - A Green Shopping Assistant**

**Group No: 3**

**Project Documentation**

**1. Project Overview**

EcoCart is a web-based platform designed to promote sustainable and eco-friendly shopping habits by offering a curated catalog of environmentally friendly products, user-submitted DIY green alternatives, and community-driven reviews. The platform allows users to explore, review, suggest, and save eco-conscious products and tutorials, creating a collaborative environment focused on sustainability.

This project was developed using Django, Python, and Bootstrap, aligning with the course objectives to build a dynamic, secure, and user-friendly web application with a strong green technology theme.

**2. Motivation and Project Uniqueness**

The rising global awareness of environmental issues motivates the need for accessible tools that encourage sustainable consumer choices. EcoCart uniquely combines a green product catalog with interactive community features such as user reviews, DIY uploads, and personalized wishlists.

The project stands out due to its:

* Exclusive focus on eco-friendly and sustainable products
* Community participation through product suggestions and DIY tutorials
* Advanced search and filter capabilities tailored to sustainability metrics
* Educational resources linked to trusted external Canadian green websites
* User activity tracking enhancing personalized experiences

**3. Tools and Technologies Used**

* **Backend**: Django framework using Python for server-side logic
* **Frontend**: Bootstrap CSS for responsive, modern UI design
* **Database**: SQLite for data storage with initial data loading
* **Development Environment**: PyCharm IDE
* **Authentication**: Django's built-in authentication system for secure login, registration, and password management
* **File Handling**: Django Media settings for user-uploaded images and documents

**4. Project Implementation Strategy**

**A. Models**

The core data structures include:

* **Product**: Stores detailed information about eco products including name, description, category, sustainability tags, and ratings
* **Review**: Captures user feedback on products with ratings and comments
* **DIY**: Stores user-uploaded tutorials with descriptions and media files
* **UserProfile**: Extends default user info with favorites and contribution stats
* **VisitLog**: Tracks user session history including pages visited and time spent
* **Category**: Classifies products by eco-related tags like compostable, reusable, solar-powered
* **WishlistItem**: Links users to their saved favorite products
* **ExternalResource**: Contains curated links to educational green websites

**B. Views**

Class-based views (CBVs) are implemented for clear separation of concerns:

* **ProductListView & ProductDetailView**: Display products with filtering and detailed information including reviews
* **ReviewCreateView**: Allows authenticated users to submit product reviews
* **DIYListView & DIYCreateView**: Show user tutorials and enable uploads
* **UserDashboardView**: Displays user-specific data such as favorites, reviews, and uploads
* **SearchView**: Implements search across products and DIYs with dropdown filters also added autocomplete for search.
* **HistoryView**: Shows user visit history using session and cookie data
* **Authentication Views**: Login, logout, registration, and password reset

**C. Forms**

User input and validation are handled through various forms:

* Search forms with text input and dropdown filters
* Review submission forms with rating and comments
* DIY upload forms for files and descriptions
* User login, registration, and password reset forms
* Wishlist management forms for adding/removing favorites

**D. User Authentication and Access Control**

* Only registered users can submit reviews, upload DIYs, and manage wishlists
* Unregistered users can browse the product catalog and DIY tutorials
* Django authentication system is utilized for secure login, logout, registration, and password recovery

**E. User History and Session Tracking**

* Sessions and cookies are employed to track user visits, page views, and activity duration
* VisitLog model persists this data for analysis and display on user dashboards
* The dashboard includes recent visits and user contribution counts

**F. Search Functionality**

* A combined search bar supports full-text search over products and DIY tutorials
* Dropdown filters allow narrowing results by category, rating, or sustainability tag
* Results are displayed with appropriate visual cues to distinguish products from DIY tutorials

**G. File Uploads**

* Users can upload DIY tutorials including images and documents
* Uploaded files are stored securely with Django's media settings
* Forms handle file validation

**H. UI/UX Design**

* Responsive design using Bootstrap ensures compatibility across devices
* A consistent eco-friendly color scheme (greens, beige, white) aligns with the project's sustainability theme
* Icons and visual cues help users easily identify categories and product types
* Navigation bars and footers provide clear access to main features and extra pages such as Contact Us and About Us

**5. Team Roles and Contributions**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Member Name** | **Student ID** | **Responsibilities** |
| **1** |  |  | Models: Product, Category Views: ProductListView, ProductDetailView Forms: Search, Wishlist |
| **2** |  |  | Models: Review, VisitLog Views: ReviewCreateView, HistoryView Forms: Review, UserLogin |
| **3** | **Anisa Hussain** | **110173274** | Models: DIY, UserProfile Views: DIYListView, DIYCreateView Forms: DIYUpload, Registration |
| **4** |  |  | Models: ExternalResource, WishlistItem Views: ExternalLinksView, UserDashboardView Forms: PasswordReset, Favorites |

**6. Project Implementation Steps**

1. **Project Setup**
   * Created Django project structure
   * Set up virtual environment and installed dependencies
   * Configured settings.py for static and media files
2. **Database Design**
   * Designed and implemented models for products, categories, reviews, DIY tutorials, etc.
   * Created relationships between models
   * Applied migrations to create database schema
3. **User Authentication**
   * Implemented user registration, login, and logout functionality
   * Created password reset functionality
   * Extended User model with UserProfile for additional information
4. **Product Management**
   * Implemented product listing with filtering and search
   * Created product detail views with reviews
   * Added sustainability tags and filtering
5. **DIY Tutorials**
   * Implemented DIY tutorial creation and listing
   * Added image upload functionality
   * Created comment system for tutorials
6. **User Dashboard**
   * Designed user dashboard with activity summary
   * Implemented wishlist functionality
   * Created visit history tracking
7. **Search and Filter**
   * Implemented global search across products and DIY tutorials
   * Added category and attribute filtering
   * Created search results page
8. **Additional Pages**
   * Created About Us page
   * Implemented Contact Us form
   * Added External Resources page with Canadian environmental websites
9. **UI/UX Design**
   * Applied Bootstrap styling for responsive design
   * Created consistent eco-friendly color scheme
   * Added icons and visual cues for better user experience
10. **Testing and Debugging**
    * Tested all functionality
    * Fixed bugs and issues
    * Ensured cross-browser compatibility

**7. Challenges and Solutions**

1. **Challenge**: URL pattern conflicts for DIY creation and detail views **Solution**: Reordered URL patterns to prioritize more specific patterns
2. **Challenge**: User visit tracking **Solution**: Created custom middleware to track user visits and page views
3. **Challenge**: Handling user uploads **Solution**: Configured Django media settings and implemented proper file validation

**8. Additional Features**

1. **Eco Points System**: Users earn points for contributing reviews and DIY tutorials
2. **Visit History Tracking**: Tracks user activity for personalized recommendations
3. **Canadian Focus**: All external resources point to Canadian environmental websites
4. **Responsive Design**: Works seamlessly on desktop and mobile devices

**9. Future Enhancements**

1. **Social Media Integration**: Allow users to share products and DIY tutorials on social media
2. **Advanced Analytics**: Provide insights on user behavior and popular products
3. **E-commerce Integration**: Add functionality to purchase products directly
4. **Sustainability Calculator**: Calculate environmental impact of product choices

**10. Conclusion**

The EcoCart project successfully fulfills the course requirements by providing a robust, interactive, and well-structured Django web application. It demonstrates practical use of models, class-based views, forms, authentication, sessions, file handling, search functionality, and responsive design. The project is built collaboratively with clear member roles and a green/environment-friendly focus, aiming to inspire sustainable consumer choices through technology.

**11. GitHub Repository**

[Link to GitHub Repository]

**12. Screenshots**

[Include screenshots of key pages and functionality]