Django

(A web development framework for Python)

Presenters:
Azhar Sikander, Ifi Okoye,
Eric Lundby, Jonathan Watson
Outline....

- Introduction to Django
- Django as MVC Design Pattern
- Django Architecture
- Django Modules
- Why Django for Web Development
- Steps to create New Project
- What Django generates
- Sample Application
Introduction – What is Django?

- Django is pronounced JANG-oh. The “D” is silent.
- “Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.”
- The web framework for perfectionists with deadlines.

Primary Focus

- Dynamic and database driven websites
- Content based websites

Example

- Websites (Washingtonpost, eBay, craigslist)
- Google AppEngine
Introduction...(Cont.)

- **History**
  - Named after famous Guitarist “Django Reinhardt”
  - Developed by Adrian Holovaty and Jacob Kaplan-moss at World Online News for efficient development
  - Open sourced in 2005
  - First Version released September 3, 2008
Django as an MVC Design Pattern

- **MVT Architecture:**
  - **Models**
    - Describes your data structure/database schema
  - **Views**
    - Controls *what* a user sees
  - **Templates**
    - **How** a user sees it
  - **Controller**
    - The Django Framework
    - URL parsing
Django Modules

- Administration interface (CRUD interface)
- Authentication system
- Comments system
- Forms handling
- Sessions
- Syndication framework (RSS and Atom Feeds)
- Caching
- Internationalization
- Localization
- Custom Middleware
Django Architecture
Why Django for Web Development

- Lets you divide code modules into logical groups to make it flexible to change
  - MVC design pattern (MVT)
- Provides auto generated web admin to ease the website administration
- Provides pre-packaged API for common user tasks
- Provides you template system to define HTML template for your web pages to avoid code duplication
  - DRY Principle
- Allows you to define what URL be for a given Function
  - Loosely Coupled Principle
- Allows you to separate business logic from the HTML
  - Separation of concerns
- Everything is in python (schema/settings)
Steps to create New Project

- Create a project
- Start an application
- Create the database (MySQL, Postgresql, SQLite)
- Define DB Settings in Settings.py
- Define your models
- Add pluggable modules
- Write your templates
- Define your views
- Create URL mapping
- Test Application
- Deploy Application (Linux, Apache, mod_Python, DB)
What Django generates...

- **MySite/**
  - `__init__.py`
  - `Manage.py`  // Script to interact with Django
  - `Settings.py`  // Config
  - `URLs.py$`  // My Site URL mapping

- **MyProject/**
  - `__init__.py`
  - `URLs.py`  // Project specific URL mapping
  - `Models.py`  // Data Models
  - `Views.py`  // Contains the call back functions
  - `Admin.py`
  - `Templates`
Show Sample Application
Questions?
(Questions are guaranteed in life; Answers aren’t.)

- For more information (Documentation, Download and News)
  - http://www.djangoproject.com/
- A Good book to learn Django
  - http://www.djangobook.com/en/1.0/
- The best place to start
- A lot of Django Pluggables available online
  - Explore at http://www.djangopluggables.com